Roteamento Multicast na Internet

Luís Henrique Maciel Kosmalski Costa
luish@gta.ufrj.br

Otto Carlos Muniz Bandeira Duarte
otto@gta.ufrj.br

Grupo de Teleinformática e Automação – GTA
Universidade Federal do Rio de Janeiro
COPPE/EE - Programa de Engenharia Elétrica
C. P. 68504 - CEP 21945-970 - Rio de Janeiro - RJ - Brasil *
Tel. +55 21 2260-5010 r.240 Fax +55 21 2562-8628

1 Princípios Básicos do IP Multicast

Muitas das novas aplicações da Internet, como vídeo-conferência, ensino à distância e TV na Internet, pertencem à categoria de comunicação de grupo, ao contrário de outras aplicações que constituem conversações ponto-a-ponto. Estas novas aplicações podem ter várias fontes e muitos receptores que podem chegar a milhões, como no caso da TV na Internet. Um serviço multicast eficiente é necessário uma vez que a utilização de múltiplos canais unicast é inviável em termos dos recursos da rede e da capacidade de processamento das estações.

O IP Multicast implementa esta funcionalidade na camada de rede, sendo composto por um modelo de serviço e gerenciamento de grupo, e por protocolos de roteamento [1]. Neste capítulo, são apresentados os conceitos fundamentais do IP Multicast.

1.1 Modelo de Serviço

O modelo de serviço IP Multicast define um grupo como uma conversação aberta, onde:

- qualquer um pode participar, sem exigência de autorização;
- uma estação pode pertencer a vários grupos diferentes, sem qualquer restrição;
- uma fonte pode enviar dados para um grupo multicast, pertencendo, ou não a este grupo;
- o grupo é dinâmico, uma estação pode entrar ou sair a qualquer momento;
- o número e a identidade dos membros do grupo não são conhecidos nem pela fonte e nem pelos receptores.

Um grupo é identificado por um endereço "IP multicast" (Seção 1.2). Alguns endereços são reservados para propósitos específicos (p. ex., identificar *todas as estações da rede*, ou *todos os roteadores*).

1.1.1 Suporte ao Multicast na Estação

Para que uma estação possa utilizar o serviço multicast, ela deve implementar o protocolo IGMP (*Internet Group Management Protocol*). Estações e roteadores devem ser capazes de lidar com um novo tipo de endereço, o endereço IP multicast. Além disso, para que uma estação suporte o serviço IP multicast, a sua interface de serviço IP deve ser estendida [2] para que:

*Apoiado por recursos da CAPES, CNPq e FAPERJ.

- esta estação possa se conectar ao grupo, ela deve reprogramar a camada rede, e, possivelmente, as camadas mais baixas, de forma a receber pacotes endereçados para grupos multicast;

- uma aplicação desta estação que se conectou a um grupo multicast, e que envia dados para este grupo, deva ser capaz de escolher se os pacotes devem ser enviados em *loopback* de forma que a aplicação receba os dados que ela produz;

- uma estação seja capaz de limitar o *escopo* dos pacotes multicast enviados. O pacote IP possui um campo *Time-To-Live* (TTL) em seu cabeçalho, que originalmente servia para limitar o tempo de vida de um pacote na rede e diminuir os efeitos de *loops* temporários do roteamento IP. Na prática, porém, o valor do campo TTL é diminuído de 1 a cada roteador atravessado pelo pacote IP, e não por um intervalo de tempo. Desta forma, no IP Multicast, o campo TTL é utilizado para controlar o alcance (em número de nós) que um pacote pode percorrer na rede a partir da fonte.

Quando uma aplicação sinaliza à camada de rede que ela deseja se conectar a um grupo multicast, o *software* de rede verifica se a estação já está conectada a este grupo. Em caso negativo, uma mensagem relatório IGMP (a ser detalhado na Seção 1.3) é enviada na rede local. Além disso, o endereço IP multicast do grupo é mapeado no endereço multicast de nível mais baixo, e a interface de rede é reprogramada de forma a aceitar pacotes multicast enviados para este grupo. É importante notar que a estação se conecta a um grupo multicast em uma interface, ou seja, uma estação com diversas interfaces de rede pode escolher a interface através da qual se conectará ao grupo multicast.

No nível da rede local, o serviço IP multicast pode tirar vantagem da tecnologia de rede utilizada. Considere por exemplo a tecnologia Ethernet. No Ethernet, são possíveis transmissões ponto-a-ponto, broadcast, e multicast. Uma faixa de endereços MAC do Ethernet foi reservada para o serviço multicast. Os endereços multicast Ethernet possuem o bit menos significativo do byte mais significativo igual a 1. O endereço multicast Ethernet é então formado pela operação lógica "ou" dos 23 bits menos significativos do endereço IP multicast com o endereço Ethernet `01.00.5E.00.00.00` (Figura 1). Por exemplo, o endereço IP multicast `224.0.2.2` é mapeado no endereço multicast Ethernet `01.00.5E.00.02.02`. Como consequência, endereços IP multicast que diferem apenas pelos 9 bits mais significativos são mapeados no mesmo endereço multicast Ethernet. Isto significa que a cada quadro recebido pela camada Ethernet, a camada de rede deve verificar se o endereço IP multicast corresponde ao endereço desejado.

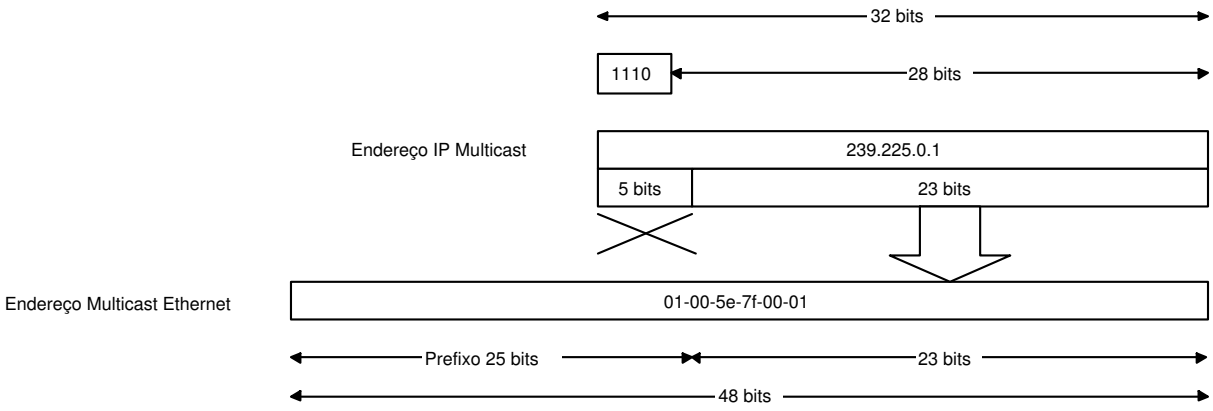


Figura 1: Mapeamento entre os endereços multicast IP e Ethernet.

Também é possível emular a transmissão multicast utilizando o broadcast Ethernet e um filtro de entrada no nível IP, esta era a solução utilizada nas primeiras placas Ethernet que não suportavam o serviço multicast.

A utilização do serviço multicast em uma rede física que não suporta a difusão (como o ATM, por exemplo) é mais complexa e menos eficiente. Algumas vezes, a solução é a utilização de um servidor que concentra o tráfego multicast e o redistribui para os receptores interessados, através de n conexões ponto-a-ponto, ou de uma conexão ponto-a-multiponto, se esta for possível (caso do ATM).

Os protocolos GARP (*Generic Attribute Registration Protocol*) e GMRP (GARP *Multicast Registration Protocol*), definidos no Padrão IEEE 802.1p [3], servem para coordenar a distribuição multicast entre pontes e comutadores. O protocolo GARP fornece a funcionalidade de "roteamento" no nível de enlace, enquanto o protocolo GMRP implementa a funcionalidade de gerenciamento de grupo, equivalente ao que o IGMP realiza no nível de rede.

1.1.2 Criação e Destruição de um Grupo Multicast

Criar um grupo multicast consiste, para a fonte de tráfego, na escolha de um endereço multicast e no envio de dados para este endereço. Após o recebimento de pacotes para um novo endereço multicast, os roteadores criam um novo estado de reenvio (*forwarding state*) e configuram uma árvore de distribuição para este grupo multicast. Se não existirem receptores interessados neste grupo, a árvore de distribuição é limitada pelo roteador correspondente ao primeiro salto na árvore (em protocolos de roteamento por fonte, p. ex., DVMRP) ou por um roteador central (no caso de protocolos de roteamento que constroem árvores centradas, p. ex., PIM-SM). No entanto, dependendo do protocolo, os roteadores podem guardar estado para este grupo, ainda que não enviem dados (o protocolo armazena estado para ramos "podados" da árvore).

A destruição de um grupo multicast consiste, simplesmente, na parada do envio de dados, pela fonte, para o endereço multicast. Como a árvore de distribuição é implementada através de *soft-states* nos roteadores, a árvore de distribuição vai desaparecer pouco a pouco, após o estouro de temporizadores (p. ex., o programa `mrouted` utiliza um temporizador de 5 minutos por *default*).

1.2 Endereçamento

Um endereço multicast não identifica uma máquina, mas um grupo, sendo dinamicamente alocado, e válido durante toda a sessão multicast.

No IPv4, um endereço multicast possui os quatro primeiros bits iguais a `1110`, o que corresponde à faixa de `224.0.0.0` a `239.255.255.255`. Esta faixa corresponde, no projeto original do endereçamento IP em classes, à Classe D. Embora a noção de Classes A, B e C tenha deixado de ser utilizada no roteamento inter-domínio [4], a Classe D continuou reservada para o IP multicast. Um endereço IP multicast possui o formato mostrado na Figura 2.

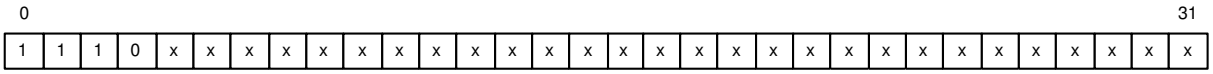


Figura 2: Formato do endereço multicast IPv4.

Em regra geral, um endereço multicast é temporário, sendo requisitado e atribuído dinamicamente. Entretanto, os endereços de `224.0.0.0` a `224.0.0.255` foram reservados pelo IANA (*Internet Assigned Numbers Authority*). Esta faixa corresponde a endereços de escopo local ao enlace (*link-local*) e portanto não devem ser encaminhados por roteadores, independente do valor

Tabela 1: Exemplos de endereços multicast reservados.

224.0.0.1	All Hosts
224.0.0.2	All Multicast Routers
224.0.0.4	All DVMRP Routers
224.0.0.5	All OSPF Routers

de seu campo TTL. Adicionalmente, alguns endereços são reservados para serviços específicos. A Tabela 1 mostra alguns exemplos desses endereços reservados.

O restante da faixa de endereços Classe D (`224.0.1.0` a `239.255.255.255`) corresponde a endereços multicast dinâmicos. O endereço multicast não é atribuído por uma entidade reguladora, como o IANA faz para as faixas de endereços unicast. Em princípio, o endereço multicast é dinamicamente escolhido pela fonte de dados. Portanto, existe um risco de que diversas fontes escolham o mesmo endereço multicast. Dependendo do escopo (definido pelo campo TTL) de cada tráfego, uma aplicação pode ser corrompida, recebendo pacotes de outras aplicações. Portanto, existe a necessidade de um esquema de alocação de endereços à parte para evitar colisões. No entanto, uma vez que não existe uma estrutura hierárquica de endereçamento, a alocação de endereços multicast pode ser complexa. Existem diferentes métodos para limitar o escopo do tráfego multicast, os principais são apresentados a seguir. Os diferentes mecanismos de alocação de endereços multicast existentes são apresentados nas Seções 1.2.4 e 1.2.5.

1.2.1 Endereços e Escopos

A faixa de endereços multicast dinâmicos (`224.0.1.0` a `239.255.255.255`) foi subdividida pelo IANA, de acordo com diferentes escopos para os grupos multicast. Os endereços de `224.0.1.0` a `238.255.255.255` podem ser utilizados por aplicações com escopo global. Já os endereços de `239.0.0.0` a `239.255.255.255` têm escopo limitado. Os endereços na faixa `239.253.0.0/16` devem ser utilizados por aplicações de escopo local ao *site*, lá os endereços `239.192.0.0/14` são reservados ao escopo local à organização.

1.2.2 Limitando o Escopo usando o Campo TTL

No IPv4, o controle do escopo dos pacotes multicast também pode ser realizado através do campo TTL (*Time To Live*) do cabeçalho IP. Numa transmissão unicast, o campo TTL é decrementado de uma unidade a cada vez que o pacote é encaminhado por um roteador. O pacote é descartado se o valor do TTL chegar a zero.

No IP Multicast, e em especial no MBone, o campo TTL é freqüentemente utilizado para limitar a distância na rede (em número de saltos) que um pacote multicast pode atingir. No entanto, nem sempre uma região pode ser definida por um determinado número de saltos. Desta forma, além do funcionamento normal do campo TTL, os roteadores são configurados com *thresholds* (valores limites) para os túneis e enlaces multicast. Quando um *threshold* está configurado, o roteador decrementa normalmente o TTL do pacote, como no unicast, mas irá descartar o pacote se o valor obtido for menor que o valor do *threshold*. Configurando-se *thresholds* consistentes nas bordas de uma região, uma estação emissora pode garantir que o seu tráfego não escapará desta região, bastando para tanto atribuir um valor de TTL dos pacotes emitidos inferior ao *threshold*. Tipicamente, os valores de TTL inicial mostrados na Tabela 2 são utilizados [5] na definição de escopos.

É importante observar que as noções de organização, região e continente não são explicitamente definidas, outros valores de TTL aparecem na literatura.

Tabela 2: Valores de TTL típicos.

Escopo	TTL inicial	*Threshold*
rede local	1	–
site	15	16
região	63	64
global	127	128

1.2.3 Escopos Administrativos

A implementação de escopos através do campo TTL do cabeçalho IP possui algumas limitações. Em alguns casos é difícil escolher um valor de TTL adequado para o escopo desejado. Por exemplo, é impossível configurar duas zonas de escopo que se sobrepõem parcialmente. Para enfrentar situações onde o escopo por TTL não é eficaz, passou-se a usar o escopo administrativo (que logo foi implementado pelo *mrouted* e na maior parte dos roteadores). O escopo administrativo define uma fronteira especificando uma faixa de endereços multicast que não serão encaminhados pelo roteador, em nenhuma das direções.

O escopo administrativo é mais flexível que o escopo por TTL (as fronteiras podem ter qualquer "formato") mas possui algumas desvantagens. Para saber a distância que um pacote vai percorrer na rede, a fonte de dados precisa conhecer todas as zonas às quais ela pertence. Além disso, uma vez que as fronteiras são bi-direcionais, duas zonas de escopo que se sobreponham, totalmente ou em parte, têm que obrigatoriamente utilizar faixas de endereços disjuntas. Desta maneira, idealmente, as zonas deveriam ser alocadas de forma hierárquica – da maior para a menor zona. Esta tarefa é no entanto complexa, uma vez que não existe uma autoridade com esta função. Além destes problemas, pode-se facilmente errar a configuração de uma fronteira, bastando para isso mal configurar, ou esquecer de configurar, um roteador. Os erros de configuração também podem ocorrer com o escopo por TTL, mas de uma maneira geral, pode-se configurar o TTL com um valor um pouco maior que o necessário e, assim, garantir o funcionamento da aplicação, com o custo de o tráfego multicast estar atingindo uma parte da rede inutilmente. Com o escopo administrativo é mais difícil de se configurar esta "margem de segurança".

Um protocolo para facilitar a configuração administrativa de escopos foi padronizado pelo IETF (*Internet Engineering Task Force*). O *Multicast Zone Announcement Protocol* (MZAP) [6] possibilita descoberta automática de zonas de escopo, assim como zonas mal configuradas.

1.2.4 Alocação Estática de Endereços Multicast

Enquanto um mecanismo de alocação dinâmica de endereços multicast era definido pelo IETF, um mecanismo estático de alocação, chamado endereçamento GLOP [7], foi proposto como solução transitória para o problema de alocação de endereços. A faixa de endereços `233/8` foi reservada pelo IANA para a alocação estática de endereços multicast [7]. A idéia básica do mecanismo é incluir o identificador de Sistema Autônomo (AS – *Autonomous System*) no endereço multicast. Assim, os 16 bits do número de AS (número que identifica de maneira única os domínios utilizados no roteamento inter-domínio) são tomados como os 16 bits do meio do endereço multicast, como mostrado na Figura 3.

Desta forma, ao Sistema Autônomo 16007 é atribuída a faixa de endereços de `233.64.7.0` a `233.64.7.255`.

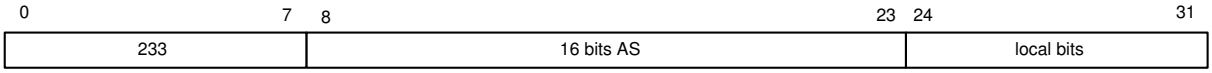


Figura 3: Formato do endereço multicast IPv4 alocado estaticamente.

1.2.5 Arquitetura de Alocação Dinâmica de Endereços Multicast

O IETF criou um grupo de trabalho específico para a elaboração de uma arquitetura de alocação de endereços multicast. O resultado do trabalho do grupo MALLOC (*Multicast-Address Allocation*) foi a arquitetura MAAA (*Multicast Address Allocation Architecture*) [8]. A arquitetura MAAA possui três camadas, que incluem um protocolo cliente-servidor (MADCAP), um protocolo intra-domínio (Multicast AAP) e um protocolo inter-domínio (MASC).

O protocolo MADCAP (*Multicast Address Dynamic Client Allocation Protocol*) [9] é um protocolo cliente-servidor que permite que as estações requisitem os serviços de alocação de endereços multicast de um servidor de endereços multicast (o servidor MADCAP). O design do protocolo é semelhante ao do protocolo DHCP (*Dynamic Host Configuration Protocol*) [10].

O protocolo Multicast AAP (*Multicast Address Allocation Protocol*) [11] é utilizado pelos servidores MADCAP para coordenar a alocação de endereços dentro de um domínio, de forma a evitar colisões (a alocação de um mesmo endereço multicast por duas aplicações diferentes).

O protocolo MASC (*Multicast Address Set Claim*) [12] se encontra no topo da hierarquia da arquitetura MAAA. O MASC é utilizado para coordenar a alocação de endereços no nível inter-domínio. O MASC serve para um nó (normalmente, um roteador) requisitar e alocar um ou mais prefixos (faixas) de endereços multicast para o domínio ao qual o nó pertence. Estes endereços serão utilizados por grupos iniciados por estações dentro de seu domínio.

O MASC utiliza o mesmo conceito de domínios do roteamento inter-domínio da Internet, ou seja, o de Sistemas Autônomos (AS), e trabalha em conjunto com o BGP (*Border Gateway Protocol*) [13] para difundir a informação de alocação de endereços. A alocação é hierárquica, domínios filhos (ou sub-domínios) escutam as faixas de endereços multicast alocadas por seus domínios pais, e selecionam sub-faixas que serão utilizadas de acordo com suas necessidades. Quando um roteador MASC percebe que não há espaço de endereçamento multicast suficiente, ele procede à requisição de uma faixa maior.

O MASC aloca as faixas de endereços de forma dinâmica, utilizando um mecanismo de escuta e pedido, com detecção de colisões. Assim, domínios-filhos escutam as faixas de endereços alocadas por seu pai, podendo selecionar sub-faixas a partir destas faixas, e devem neste caso propagar as faixas selecionadas a seus "irmãos", tornando público o requerimento destas faixas. Os nós que estão em processo de requisição de uma faixa devem esperar um determinado tempo, antes de considerarem-la alocada, para detectar possíveis colisões com as faixas alocadas por seus irmãos. Após este período, as faixas alocadas com sucesso devem ser informadas ao servidor MAAS do domínio, e aos outros domínios, através do protocolo BGP. Para tanto, são utilizadas "rotas de grupo" (*group routes*) do BGP. Estas rotas de grupo do BGP são utilizadas pelo protocolo BGMP (*Border Gateway Multicast Protocol*) [14] para a construção de árvores multicast inter-domínio (Seção 3.2). Após este processo, os servidores MAAS podem atribuir endereços multicast da faixa alocada a grupos iniciados dentro de seu domínio.

A Figura 4 ilustra a alocação hierárquica feita pelo MASC. Os domínios A, D e E são provedores de *backbone*. Os domínios B e C são provedores regionais, clientes de A, e possuem os provedores F e G como clientes, respectivamente. Suponha que B deseja obter uma faixa de endereços para seu domínio, e que o Domínio A já tenha obtido a faixa de endereços `224.0.0.16`, utilizando o MASC.

Os domínios B e C são filhos de A, que anuncia sua faixa de endereços, `224.0.0.16`, a todos

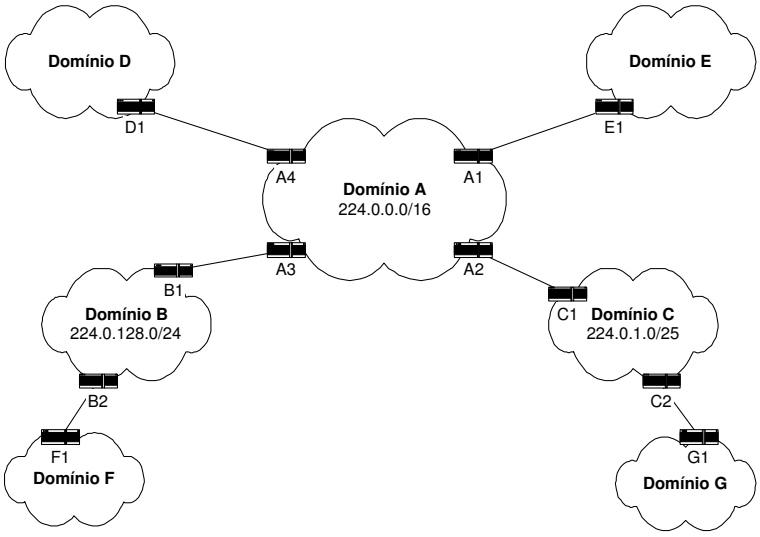


Figura 4: Alocação hierárquica de endereços pelo MASC.

os seus filhos. Suponha que B requeira a sub-faixa `224.0.1.0/24` da faixa de A. B informa seu domínio pai, A, e outros domínios-irmãos que estejam diretamente conectados a B, desta requisição. O domínio A por sua vez propaga este pedido a todos os outros domínios-filhos.

No caso em que algum dos domínios-irmãos já tenha alocado a faixa pedida, ou parte desta, este envia um anúncio de colisão. Este é o caso do domínio C, que possui a faixa `224.0.1.0/25`. Ao ouvir o anúncio de colisão, B deve tentar uma nova faixa, por exemplo `224.0.128.0/24`. Em seguida, B deve esperar por anúncios de colisão durante um determinado intervalo de tempo, antes de considerar que a faixa tenha sido alocada com sucesso. Se nenhuma colisão ocorrer, B comunica a faixa de endereços obtida a seus servidores MAAS locais, e a outros domínios, utilizando rotas de grupo BGP.

Como será visto em mais detalhes na Seção 3.2, quando um domínio X anuncia uma rota de grupo BGP, para a faixa multicast F, isto significa que X pode ser utilizado como rota para chegar ao domínio-raiz dos grupos dentro da faixa F. Por exemplo, o roteador de borda $B1$ anuncia a rota de grupo correspondente à faixa `224.0.128.0/24` ao roteador $A3$ no domínio A. Uma vez que todos os roteadores de borda de um domínio são ligados por uma malha de conexões BGP internas, a rota de grupo de $B1$ é passada para $A1$, $A2$ e $A4$. Como o domínio A só recebeu uma rota para este prefixo, a rota por B é a escolhida. Se houvesse múltiplas rotas, a escolha seria realizada através dos atributos das rotas, como feito pelo BGP para rotas unicast.

As rotas de grupo são armazenadas pelos roteadores BGP dentro de uma tabela chamada G-RIB (*Group - Route Information Base*). A rota de grupo de $B1$ é armazenada por $A3$ em sua G-RIB como (`224.0.128.0/24`, $B1$), indicando que o roteador $B1$ é o próximo salto para chegar ao domínio raiz para os grupos dentro de `224.0.128.0/24`. Os roteadores $A1$, $A2$ e $A4$ armazenam (`224.0.128.0/24`, $A3$) em suas G-RIBs, uma vez que $A3$ é o seu próximo salto correspondente.

A agregação de rotas de grupo funciona da mesma forma que para as rotas unicast no BGP. Por exemplo, uma vez que a faixa de endereços de A, `224.0.0.0/16`, engloba a faixa de endereços da rota de $B1$, `224.0.128.0/24`, os roteadores de borda de A não anunciam a rota de $B1$ a outros domínios, mas apenas a rota agregada. No exemplo acima, o roteador $A1$ anuncia a rota de grupo (`224.0.0.0/16`, $A1$) ao roteador $E1$ no domínio E. Desta forma, a informação de alocação de endereços multicast pelo protocolo MASC é propagada a todos os outros domínios.

Políticas de transmissão para o tráfego multicast podem ser implementadas, da mesma forma

que para o tráfego unicast, pela propagação seletiva dos anúncios de rotas de grupo pelo BGP. Por exemplo, um domínio pode anunciar rotas de grupo apenas para as faixas de endereços alocadas por nós dentro de seu próprio domínio, ou por domínios clientes. Desta forma, ele pode evitar a utilização de seus recursos por tráfegos encaminhados para um domínio-raiz de terceiros.

A separação da alocação de endereços multicast em dois níveis, inter-domínio e intra-domínio, permite com que endereços multicast sejam alocados rapidamente, internamente a um domínio, da mesma forma que endereços unicast são rapidamente alocados dentro de uma rede, enquanto que a alocação de prefixos unicast para um domínio é mais lenta.

1.3 Gerenciamento de Grupo - IGMP/MLD

O *Internet Group Management Protocol* (IGMP) é um protocolo de nível 3 (como o ICMP). No IPv6, o protocolo responsável pelo gerenciamento de grupos multicast é o MLD (*Multicast Listener Discovery*). O IGMP é responsável por informar o roteador multicast local da presença de estações interessadas em escutar um determinado grupo multicast. Diferentes versões do protocolo IGMP existem:

- IGMP versão 1: especificado em [15].
- IGMP versão 2: especificado em [16].
  A maior diferença do IGMPv2 em relação ao IGMPv1 é a introdução do mecanismo de "fast leave".
- IGMP versão 3: especificado em [17].
  A versão 3 adiciona a possibilidade de filtragem de tráfego por fonte.

A operação do protocolo IGMP pode ser dividida em duas partes: o lado estação e o lado roteador. Quando uma estação se conecta pela primeira vez a um grupo multicast G, ela programa sua interface de rede para receber o tráfego multicast desejado e envia uma mensagem IGMP *Join* na rede local. Esta mensagem informa o(s) roteador(es) locais que existe um receptor interessado no grupo multicast G.

Para conhecer a identidade dos grupos multicast para os quais existe interesse nas suas redes locais, o roteador multicast local envia periodicamente mensagens "QUERY", interrogando as estações conectadas à rede local sobre a presença de "ouvintes" para algum grupo multicast. Cada estação que deseje escutar um grupo responde à interrogação com uma mensagem relatório ("REPORT"), onde são listados os endereços multicast dos grupos nos quais a estação está interessada. Um mecanismo de cancelamento de resposta é utilizado, para evitar a implosão do roteador multicast. Um receptor não envia o seu relatório imediatamente após a recepção de um QUERY, mas apenas após um intervalo de tempo aleatório, durante o qual ele escuta os relatórios emitidos por outras estações. Se os grupos nos quais o receptor tem interesse forem pedidos por outras estações, ele pode suprimir o envio do seu relatório.

Quando a aplicação correspondente ao grupo multicast G termina, e a estação não tem mais interesse em receber o tráfego endereçado para G, a estação reprograma sua interface de rede para não mais repassar o tráfego multicast para a camada superior, e pára de enviar mensagens IGMP.

No IGMPv1, quando todas as estações que estavam conectadas ao grupo G na rede local deixam de escutar esse grupo, o tráfego para o grupo G continua a ser enviado por um intervalo de tempo, equivalente ao *timeout* (tempo de validade) do estado G no roteador. Para diminuir a latência de desconexão do grupo, no IGMPv2 foi introduzido um novo tipo de mensagem, *Leave*, que permite a desconexão explícita do grupo multicast. Além disso, um conjunto de regras de processamento garante que um receptor que deixa o grupo não desconectará os outros.

O IGMP é um protocolo de sinalização que se restringe ao diálogo local entre as estações receptoras e o roteador multicast local (ou roteador de primeiro salto). O escopo do IGMP é local. A criação da árvore de distribuição multicast é independente do IGMP, sendo responsabilidade dos roteadores multicast no escopo de longa distância (*wide-area*), que executam um protocolo de roteamento multicast.

1.3.1 O Protocolo IGMPv3

A Versão 3 do IGMP acrescenta o suporte a "filtragem de fontes", ou seja, as estações têm a possibilidade de anunciar o interesse na recepção de pacotes enviados para um endereço de grupo multicast, "apenas por determinadas fontes", ou "por qualquer fonte, exceto determinadas fontes". Essa informação pode ser utilizada pelos protocolos de roteamento multicast para evitar o envio de pacotes de determinadas fontes multicast para redes onde não existem receptores interessados [17]. Para permitir a configuração dos filtros no IGMP, uma modificação da API socket é necessária. Desta forma, um pedido IGMPv3 possui o seguinte formato:

```
IPMulticastListen (socket, interface, mcast-address, filter mode, source list)
```

onde o "filter mode" pode ser INCLUDE ou EXCLUDE. Por exemplo, para especificar que um receptor deseja escutar o grupo multicast G e receber apenas os dados enviados pelas fontes S1 e S2, o pedido terá a seguinte forma:

```
IPMulticastListen (socket, interface, G, INCLUDE,{S1, S2})
```

Se por outro lado o receptor deseja ouvir o grupo G mas receber dados enviados por todas as fontes, exceto S1 e S2, o pedido será:

```
IPMulticastListen (socket, interface, G, EXCLUDE, {S1, S2})
```

As operações equivalentes ao $join(*, G)$ e $leave(*, G)$ das versões anteriores do IGMP são equivalentes a:

```
Join = IPMulticastListen (socket, interface, G, EXCLUDE, {})
```

```
Leave = IPMulticastListen (socket, interface, G, INCLUDE, {})
```

2 Roteamento Multicast Intra-domínio

O protocolo IGMP é restrito ao diálogo local entre os receptores e o primeiro roteador multicast. Para que o tráfego multicast seja encaminhado através de vários saltos, a criação de uma árvore de distribuição é necessária, sendo responsabilidade do protocolo de roteamento. Este capítulo apresenta os principais protocolos de roteamento IP multicast, evidenciando sua evolução desde o DVMRP, o primeiro protocolo proposto com o modelo de serviço IP Multicast.

2.1 Os Diferentes Algoritmos de Roteamento Multicast

Existem diferentes algoritmos para a criação da árvore de distribuição multicast. No entanto, de uma maneira geral, os protocolos de roteamento multicast mais difundidos podem construir dois tipos de árvore de distribuição: árvores por fonte (*source-based trees*, ou *source-specific trees*) ou árvores compartilhadas (ou centradas – *center-based trees*). A diferença básica entre as duas

árvores de distribuição é que para cada fonte de dados multicast é necessária a criação e manutenção de uma árvore por fonte, enquanto que a árvore compartilhada é utilizada por várias fontes. Como conseqüência, a raiz da árvore por fonte é o próprio nó fonte de tráfego, enquanto que a árvore compartilhada tem por raiz um nó arbitrário da rede (idealmente localizado no "centro" da rede). Antes de descrever os diferentes protocolos de roteamento, é apresentada a modelagem mais freqüente para o problema de roteamento multicast. Em seguida, são apresentados os principais algoritmos propostos na literatura, e que serviram de base para a implementação dos protocolos de roteamento multicast Internet.

2.1.1 O Problema de Roteamento Multicast

O problema do roteamento multicast pode ser modelado da seguinte forma. A rede pode ser modelada por um *grafo direcionado*, G, consistindo de um conjunto V de vértices (nós) e de um conjunto E de enlaces [18, 19]. Um enlace direcionado do grafo G, entre os nós u e v é representado pelo par (u, v). Considere que M é o conjunto dos nós do grupo multicast, incluindo os nós fonte (M é um sub-conjunto de V). O problema de roteamento multicast consiste em descobrir uma ou mais topologias de interconexão, árvores, sub-conjuntos de G, que incluem todos os nós em M. Quando uma única árvore é construída, independente do nó fonte, esta é uma topologia de interconexão compartilhada, ou árvore compartilhada. Quando várias topologias são construídas, uma por fonte, a solução para o problema é um conjunto de topologias de interconexão direcionadas pela fonte, ou árvores por fonte. Nas seções a seguir são apresentados exemplos de algoritmos pertencentes às duas categorias de soluções.

É importante observar que o modelo apresentado não considera a presença de enlaces simétricos. Na prática, os enlaces de transmissão podem ser bi-direcionais e possuir características semelhantes nas duas direções, o que significa que o grafo G possui diversos enlaces não-direcionados.

2.1.2 O Algoritmo de Inundação

No algoritmo de inundação, ao receber um pacote multicast, o nó verifica se esta é a primeira vez que o pacote é recebido. Se este é o caso, o pacote é enviado em cada uma das interfaces de saída, exceto aquela pela qual o pacote foi recebido. Caso contrário o pacote é descartado.

A dificuldade está no teste de "primeira recepção" do pacote. Uma solução seria o nó gravar todos os pacotes recebidos até então. Outra possibilidade seria cada pacote carregar a lista dos nós atravessados. Por exemplo, o protocolo OSPF (*Open Shortest Path First*) [20] utiliza um algoritmo de inundação onde o roteador compara a data (um número de controle do banco de dados) do pacote recebido com a data de modificação de seu próprio banco de dados.

Embora este algoritmo seja simples e robusto, sua implementação é inviabilizada pelo consumo de memória e recursos da rede. No entanto, a inundação serviu de base para algoritmos mais complexos utilizados em alguns dos protocolos de roteamento multicast implementados.

2.1.3 Árvores de Cobertura

Outra solução para o problema de roteamento multicast consiste na construção de uma árvore de cobertura (*spanning tree*). A árvore de cobertura consiste num sub-grafo da topologia da rede incluindo todos os nós em M, sem ciclos fechados. Pode-se adicionar o objetivo de custo mínimo ao problema, onde o custo total é igual à soma dos custos de cada enlace utilizado na árvore (freqüentemente, utiliza-se um custo unitário para cada enlace, e o custo da árvore é igual ao número de enlaces da árvore). Este tipo de árvore de cobertura de custo mínimo é conhecido como árvores de Steiner. O problema de árvores de Steiner em redes é NP-completo no caso geral. Adicionalmente, no caso mais geral, pode ser associado um custo c_{uv} a cada enlace.

Se o custo mínimo não for tomado como objetivo, um algoritmo simples existe: (1) seleciona-se um núcleo (nó raiz) e (2) forma-se a árvore com os enlaces utilizados nos caminhos mais curtos

(*shortest paths*) entre o nó raiz e todos os outros nós do grupo. Esta é a idéia básica do algoritmo RPF (*Reverse Path Forwarding*), descrito na seção a seguir.

Ao contrário do algoritmo de inundação, numa árvore de cobertura alguns enlaces da rede podem não ser utilizados na distribuição multicast.

2.1.4 Árvores RPF

A construção de uma árvore de cobertura por fonte (na realidade, por sub-rede fonte de dados) é mais simples que uma árvore de cobertura para todas as fontes e receptores. O primeiro algoritmo para construir este tipo de árvore, e que mais tarde deu origem ao RPF (*Reverse Path Forwarding*) era conhecido por *Reverse Path Broadcasting* (RPB).

O algoritmo RPB é simples. Ao receber um pacote da fonte S, um roteador R verifica se o pacote foi recebido pela interface de saída que ele usaria para enviar dados a S, ou seja, se o pacote chegou pelo caminho *reverso* entre a fonte S e o roteador R. Em caso afirmativo, o roteador reenvia o pacote em todas as interfaces de rede, exceto a interface por onde o pacote chegou. Senão, o pacote é descartado. Este teste é conhecido como RPF *check*. A interface pela qual o roteador espera receber pacotes de uma determinada fonte é referenciada como o enlace "pai". Os enlaces pelos quais o pacote é enviado são conhecidos como "filhos".

Este algoritmo básico foi melhorado para reduzir a duplicação desnecessária de pacotes, dando origem ao algoritmo *Reverse Path Forwarding* (RPF). Para tanto, é preciso que o roteador que recebe um pacote multicast seja capaz de determinar se o roteador no enlace "filho" o considera como estando no caminho mais curto para a fonte. Se este for o caso, o pacote é enviado para este roteador vizinho. Se não for o caso, o pacote não é enviado para este vizinho, uma vez que este iria fatalmente descartá-lo por não estar chegando através do enlace "pai" para aquele determinado par (fonte,grupo).

A informação necessária para tomar esta decisão de envio no sentido "descendente" do fluxo é relativamente fácil de obter de um protocolo de roteamento baseado no estado do enlace (*link-state*), uma vez que cada roteador mantém uma base de dados da topologia para todo o domínio de roteamento. Se um protocolo de roteamento do tipo vetor-distância for utilizado, um vizinho pode avisar seu salto anterior de um determinado par $(fonte, grupo)$ através de suas mensagens de atualização de roteamento ou eliminar a rota reversamente. Qualquer das duas técnicas permite a um roteador no caminho "ascendente" do fluxo determinar se o roteador vizinho o considera no caminho mais curto para determinado par $(fonte, grupo)$.

A principal vantagem do RPF é sua relativa eficiência e facilidade de implementação. O algoritmo não exige que um roteador tenha conhecimento da árvore de distribuição inteira nem requer um mecanismo especial para parar o processo de envio, como a inundação. Além disso, o RPF garante o envio eficiente pois os pacotes multicast sempre seguem o caminho mais curto da fonte de dados para o grupo de destino. Por outro lado, uma grande desvantagem é a distribuição do tráfego para todos os nós da rede, mesmo aqueles que não têm interesse neste tráfego. Para reduzir este problema, uma variante do RPF onde são utilizadas podas foi proposta. O tipo de árvore construída é chamada TBT (*Truncated Broadcast Tree*).

O primeiro passo consiste na inundação da rede (*flood*), onde todos os roteadores recebem todos os pacotes multicast. Na segunda etapa, um roteador folha que recebe um pacote multicast para o qual ele não possui receptor interessado envia uma mensagem de poda na direção contrária do fluxo. Passo a passo, os roteadores intermediários são informados da ausência de receptores multicast em partes da árvore, e evitam o envio de pacotes multicast para estas partes da rede.

Esta variante do RPF possui a vantagem de eliminar o tráfego multicast em partes da rede onde ele não é necessário, mas por outro lado, ainda necessita da inundação periódica de toda a rede, para que seja possível a conexão à árvore de novos receptores. Uma vez que os membros de um grupo e a topologia da rede estão mudando dinamicamente, o estado podado de uma árvore de envio multicast deve ser atualizado em intervalos regulares. Periodicamente, a informação de

poda é removida da memória de todos os roteadores e o próximo pacote para o par (fonte,grupo) é enviado para todos os roteadores nas folhas da árvore. Isto resulta em uma nova rajada de mensagens de poda, permitindo a árvore multicast se adaptar às necessidades da rede. Além disso, os roteadores que estão em partes da rede onde não há receptores armazenam, de qualquer forma, estado para todos os grupos multicast, o estado de poda. Os protocolos DVMRP e PIM-DM utilizam este algoritmo (conhecido como "*flood and prune*").

2.1.5 Árvores Centradas

Nos algoritmos de árvores centradas (ou árvores compartilhadas) a árvore de distribuição multicast é construída a partir de um nó central da rede. Os receptores (roteadores multicast de último salto) enviam os pedidos de conexão ao grupo para este roteador central (ou roteador *core*). Cada roteador atravessado pelo pedido de conexão armazena a interface pela qual o pedido chegou para construir a árvore de distribuição.

O roteador *core* é utilizado por todas as fontes que enviam para o grupo multicast, por isso a árvore de distribuição é chamada compartilhada. O tráfego multicast para cada grupo é enviado e recebido pela mesma árvore de distribuição, não importando a fonte. O protocolo CBT (Seção 2.5) foi o primeiro a utilizar árvores multicast centradas.

2.2 A Implantação do Serviço Multicast na Internet

O serviço IP Multicast começou a ser implantado na Internet no início da década de 90. A implantação experimental do multicast foi realizada através de uma rede virtual mundial, o MBone (*Multicast backBone*) [21].

O MBone é uma rede virtual, ou *overlay*, construído sobre a topologia Internet. O *overlay* é necessário uma vez que apenas um sub-conjunto dos roteadores da Internet implementa o serviço multicast. Portanto, o MBone é constituído por túneis que interconectam os roteadores multicast. Todos os pacotes multicast são encapsulados em pacotes unicast (utilizando encapsulamento IP dentro de IP). Desta forma o pacote pode ser encaminhado por roteadores unicast.

Em cima da rede virtual MBone é executado o protocolo de roteamento DVMRP (descrito na seção a seguir). A utilização de túneis possibilitou a rápida implementação de uma infra-estrutura multicast experimental, mas possui algumas ineficiências, como por exemplo o fato de um pacote atravessar o mesmo enlace físico mais de uma vez, se diferentes túneis multicast compartilharem este enlace.

2.3 O Protocolo DVMRP

O DVMRP (*Distance Vector Multicast Routing Protocol*) [22] foi o primeiro protocolo utilizado no MBone (Multicast Backbone), a rede virtual criada para desenvolver o IP Multicast. O DVMRP utiliza vetores de distância (como o protocolo de roteamento unicast RIP [23]) para construir rotas unicast para cada destino (neste caso, fonte de dados multicast). A distância é expressa em número de saltos, como no RIP. A árvore multicast é criada utilizando o algoritmo RPF (*Reverse Path Forwarding*) e as tabelas de roteamento unicast construídas pelo DVMRP.

Quando um roteador DVMRP recebe dados de uma determinada fonte, ele verifica se os dados chegaram pela interface que ele utilizaria para ir à fonte (caminho reverso) e então reenvia estes dados em todas suas interfaces de saída, inundando a rede. Roteadores que não estão interessados no fluxo de dados enviam mensagens de poda ao roteador de onde receberam os dados. Este mecanismo é conhecido por inundação-e-poda (*flood-and-prune*). A descoberta de fontes ativas se faz através da recepção dos dados, por isso a inundação da rede é feita periodicamente. Isto inviabiliza a utilização do DVMRP em grande escala.

Durante a implantação inicial do DVMRP no MBone, não havia podas de ramos da árvore, e as transmissões eram limitadas apenas pelo campo TTL (Seção 1.2.2). A partir de 1993, todas as

implementações do DVMRP passaram a utilizar a versão com podas do algoritmo RPF. Ainda assim, alguns pacotes inundam regularmente toda a rede MBone, limitando-se apenas pelo campo TTL e por valores limite administrativamente configurados (Seção 1.2.3). A inundação periódica é necessária para a descoberta de novos receptores.

2.3.1 Funcionamento Básico do DVMRP

As portas de um roteador DVMRP podem ser uma interface física diretamente conectada a uma sub-rede ou um túnel para outra ilha multicast. Todas as interfaces são configuradas com uma métrica que indica o custo para determinada porta e um valor TTL que limita o escopo de uma transmissão multicast. Além disso, cada interface túnel precisa ser configurada com dois parâmetros adicionais: o endereço IP da interface local do roteador e o endereço IP da interface do roteador remoto.

Um roteador multicast só irá enviar um pacote através de uma interface se o campo TTL no cabeçalho do pacote for maior que o TTL associado à interface. De acordo com o RPM (*Reverse Path Multicasting*, algoritmo implementado pelo DVMRP), o primeiro datagrama para cada par (fonte,grupo) é enviado através da rede inteira (limitado apenas pelo campo TTL do pacote). O datagrama inicial é enviado para todos os roteadores folha, que transmitem mensagens de poda de volta para a fonte caso não existam membros do grupo conectados às sub-redes nas folhas. As mensagens de poda criam uma árvore específica com o caminho mais curto para cada fonte.

O DVMRP possui um mecanismo para que um ramo podado possa ser "enxertado" de volta rapidamente. O roteador que havia enviado a mensagem de poda pode enviar uma mensagem para a eliminação desta tão logo descubra membros para um determinado (fonte,grupo) numa sub-rede conectada a si. As mensagens de enxerto (*graft*) são enviadas na direção da fonte, permitindo que ramos anteriormente podados sejam reativados.

Quando houver mais de um roteador DVMRP em uma sub-rede, um deles é escolhido como o roteador designado (*Designated Router* – DR), ficando responsável pelo envio de mensagens ao IGMP para a descoberta de membros dos grupos. O roteador com o menor endereço IP é escolhido como o DR, salvo a condição em que os roteadores possuem diferentes métricas para a fonte, caso em que o DR será o roteador com a menor métrica.

Uma vez que o DVMRP foi desenvolvido para rotear tráfego multicast e não unicast, o roteador em geral vai executar um processo para o envio multicast e outro para o unicast. O processo DVMRP periodicamente troca mensagens de atualização das tabelas de roteamento com vizinhos capazes de rotear multicast. Estas atualizações são independentes das geradas pelo protocolo utilizado para o roteamento unicast.

O DVMRP utiliza-se de uma tabela de roteamento, que não contém informação sobre os membros dos grupos multicast e de uma tabela de envio. A tabela de roteamento contém como elementos principais a sub-rede fonte, ou seja, o endereço de uma sub-rede que contenha estações gerando tráfego multicast ; a máscara para esta sub-rede; o roteador anterior na direção da sub-rede fonte e o campo TTL, utilizado para o gerenciamento da tabela significando o número de segundos até que uma entrada seja retirada. A tabela de envio contém os seguintes campos: sub-rede fonte, neste caso a sub-rede contendo estações gerando datagramas multicast endereçados a grupos específicos; o grupo multicast, endereço IP Classe D para o qual datagramas são endereçados, uma sub-rede fonte pode conter fontes para vários grupos multicast; um campo porta de entrada para cada par (fonte,grupo); e um campo portas de saída, através das quais datagramas destinados a um (fonte,grupo) são enviados.

A Figura 5 ilustra o funcionamento do protocolo de roteamento "unicast" do DVMRP. A construção da árvore TBT (*Truncated Broadcast Tree*) se dá de forma análoga ao protocolo RIP (*Route Information Protocol*) [23], ou seja, através da troca de vetores de distância. A Figura 5 mostra a construção da árvore TBT para a Rede $N1$ (ou seja, considerando que a Rede $N1$ é fonte do multicast, e as Redes $N2$ e $N3$ receptores). A árvore TBT é uma árvore reversa,

portanto, a Rede $N1$ é o "destino" para o qual se construirão rotas unicast.

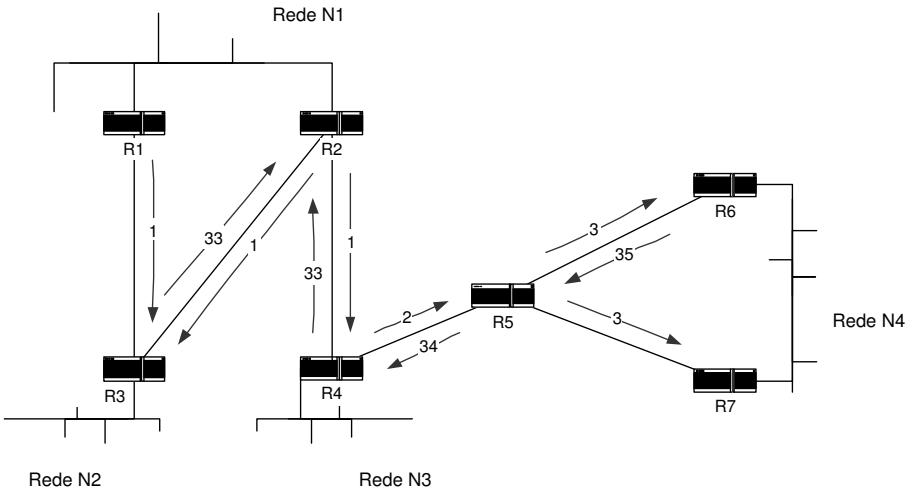


Figura 5: Construção da árvore DVMRP.

Os roteadores $R1$ e $R2$ estão diretamente conectados à $N1$, portanto $R1$ e $R2$ anunciam a seus vizinhos uma rota para $N1$ com comprimento de 1 salto. O roteador $R4$ recebe apenas um anúncio de rota para $N1$, portanto seu próximo salto para $N1$ é $R2$. Para comunicar $R2$ que $R4$ o utilizará como pai na árvore multicast, o DVMRP utiliza um mecanismo de "poison-reverse" especial. O roteador $R4$ envia um vetor de distância com métrica igual a (32 + *métrica recebida do pai*) a $R2$ (32 é o valor "infinito" no DVMRP). Desta forma, $R2$ marca a interface que o conecta a $R4$ como interface filha na árvore de distribuição de $N1$, construindo um ramo da árvore. O roteador $R3$ recebe dois anúncios de rota para $N1$ de mesmo comprimento, para escolher uma única rota, $R3$ escolhe a rota anunciada pelo vizinho de endereço IP mais baixo (suponha que seja $R2$). O roteador $R3$ envia então um vetor com distância 33 para $R2$, construindo mais um ramo da árvore TBT. Na próxima etapa, os roteadores $R3$ e $R4$ enviam vetores de distância a seus demais vizinhos (com a métrica da rota que eles possuem, acrescida de 1 salto, exatamente como no funcionamento do RIP). Desta forma, $R5$ aprende que possui uma rota para $N1$, passando por $R4$, com dois saltos. $R5$ envia o "poison-reverse" correspondente a $R4$. Finalmente, $R6$ e $R7$ recebem o vetor de distância de $R5$. Neste cenário, $R6$ e $R7$ estão na mesma rede, e portanto apenas um dos roteadores deve se conectar à árvore, o roteador designado (DR). No DVMRP, o roteador com endereço IP mais baixo é escolhido geralmente como DR (suponha que seja $R6$ neste cenário).

Considere agora um nó na rede $N1$ que envia dados para o grupo multicast $G1$. Existem receptores interessados em $G1$ nas redes $N3$ e $N4$ (Figura 6). Inicialmente, a única informação de roteamento que os nós possuem é o estado deixado pelo protocolo "unicast" do DVMRP, ou seja, uma árvore TBT que cobre *todas* as sub-redes da topologia. Portanto, no início os dados enviados pela fonte $S1$ irão inundar toda a rede. Como na rede $N2$ não existem receptores interessados no grupo $G1$, o roteador $R3$ envia uma mensagem de poda (*prune*) ao seu roteador pai ($R2$), de forma que o tráfego cessa de fluir para $N2$. A ausência de receptores interessados no grupo é detectada, em vez de sua presença. Por isso, é necessário que o processo de inundação seja repetido periodicamente (os estados de poda são voláteis), pois esta é a única forma de anunciar a existência de novas fontes/grupos aos eventuais receptores em redes podadas, como $N2$.

A utilização de uma árvore TBT pelo DVMRP tem a vantagem de restringir a inundação inicial do tráfego multicast aos ramos desta árvore. No entanto, redes onde não existem receptores são periodicamente inundadas. Além disso, o DVMRP implementa seu próprio "roteamento

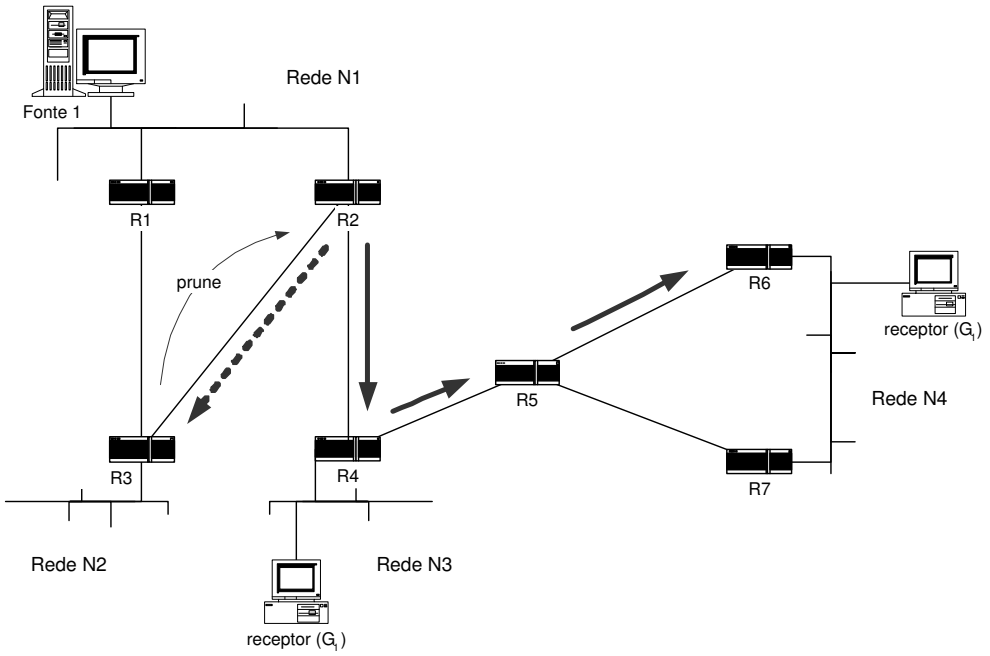


Figura 6: Envio de dados no DVMRP.

unicast", em vez de utilizar as informações do protocolo unicast em uso (como faz o protocolo PIM). Ainda com relação ao roteamento unicast, o DVMRP, por utilizar vetores de distância, sofre dos mesmos problemas de convergência lenta que seu equivalente unicast, o RIP [24].

2.4 O Protocolo MOSPF

O *Multicast Open Shortest Path First* (MOSPF) [25] é uma extensão do protocolo de roteamento unicast OSPF (*Open Shortest Path First*) [20]. Este é um protocolo projetado especificamente para distribuir informação de topologia unicast em um único Sistema Autônomo (AS). O OSPF utiliza mensagens de "estado do enlace" (*Link State Advertisement* – LSA) que são trocadas por todos os roteadores da rede, de forma que cada um é capaz de construir um mapa atualizado da topologia da rede [24]. Com esta informação, cada roteador é capaz de calcular o caminho mais curto entre dois nós quaisquer da rede, utilizando o algoritmo de Dijkstra.

Os roteadores MOSPF mantêm uma imagem da topologia atual da rede através do protocolo OSPF. O MOSPF é construído sobre o OSPF versão 2 [25], e mantém a compatibilidade com este. O MOSPF, diferentemente do DVMRP, não provê suporte para túneis. O MOSPF estende as mensagens de atualização do estado do enlace do OSPF com a informação de grupos multicast ativos. Cada roteador anuncia a presença de receptores interessados em determinado grupo multicast conectados a alguma de suas sub-redes. Desta forma, o MOSPF pode construir uma árvore de distribuição multicast baseada nos caminhos mais curtos para cada receptor (o caminho entre a fonte e o receptor é o mesmo caminho unicast). Este tipo de árvore de distribuição é chamado de árvore de caminhos mais curtos, ou árvore SPT (*Shortest-Path Tree*). A utilização das mensagens de estado de enlace dispensa a utilização da inundação periódica de dados feita pelo DVMRP, mas por outro lado inviabiliza a utilização do MOSPF em redes muito grandes.

O roteamento no MOSPF[26] pode ser dividido em intra-área e inter-área, em termos da área OSPF em questão. A Figura 7 mostra os principais elementos do roteamento intra-área. Os roteadores MOSPF utilizam-se do IGMP para obter informação sobre os grupos e seus membros. Assim, os roteadores mantêm uma base de dados local com os grupos e respectivas listas de

membros, diretamente conectados a uma sub-rede. Em qualquer sub-rede, as consultas ao IGMP são realizadas por um único roteador, o *Designated Router* (DR), responsável também por escutar as mensagens de entrada de estações em grupos geradas pelo IGMP. Num ambiente contendo roteadores MOSPF e OSPF, um roteador MOSPF deve ser o DR. No exemplo da Figura 7, o roteador $R7$ anuncia aos outros roteadores da área que ele possui receptores para os grupos multicast G_A e G_B, através da inundação da rede com LSAs listando estes grupos. Da mesma forma, o roteador $R9$ inunda a rede com LSAs anunciando a existência de receptores interessados no grupo G_A.

O caminho mais curto para cada $(fonte, grupo)$ é construído sob demanda quando o roteador recebe o primeiro datagrama para este par específico. Após esta chegada, a árvore de distribuição pode ser construída através do algoritmo de Dijkstra. Desta forma, a fonte $S1$ pode construir a árvore SPT mostrada na Figura 7. Em seguida, as informações de grupo armazenadas podem ser utilizadas para a poda dos ramos que não levam a sub-redes contendo membros do grupo específico. Para enviar um datagrama multicast para os membros de um grupo no sentido descendente do fluxo, o roteador deve determinar sua posição na árvore.

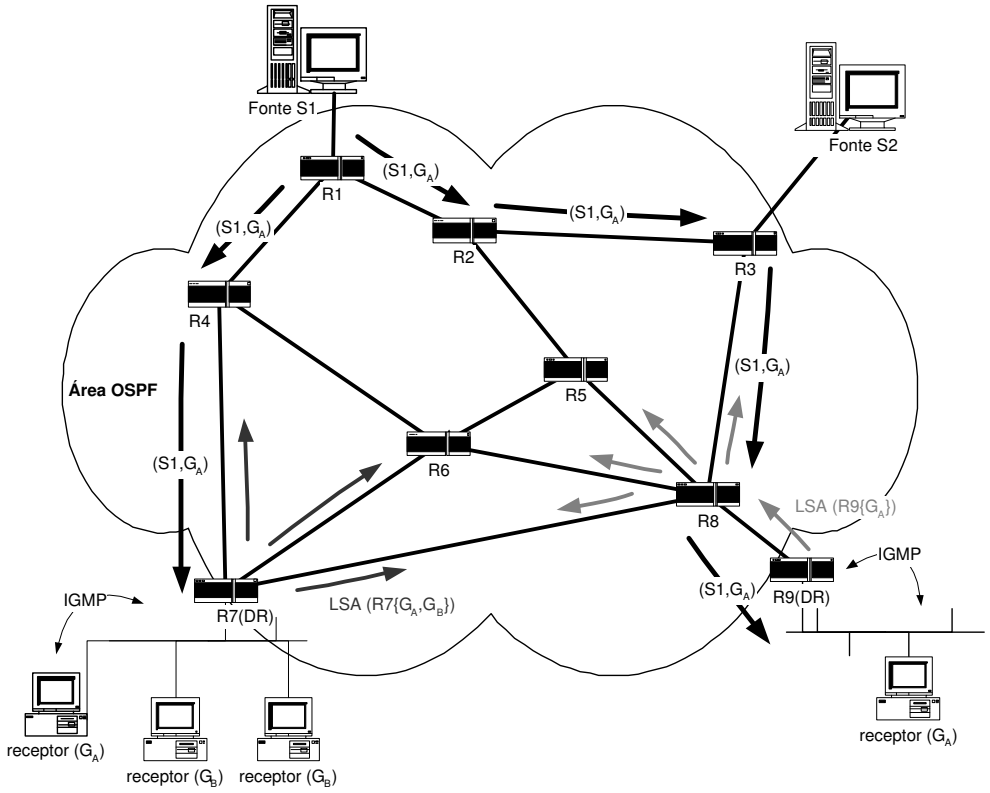


Figura 7: Funcionamento do MOSPF intra-área.

Para um determinado datagrama, todos os roteadores em uma mesma área OSPF calculam a mesma árvore de envio de caminho mais curto. Diferentemente do OSPF, o MOSPF não suporta o conceito de roteamento com vários caminhos de mesmo custo. Em contraste com o DVMRP, o primeiro datagrama multicast não precisa ser enviado para todos os roteadores na área. O cálculo sob demanda possui a vantagem de espalhar no tempo o cálculo das rotas, resultando em um menor impacto aos roteadores participantes.

Cada roteador MOSPF possui uma tabela de *cache* de envio, contendo como informação principal os endereços de grupo de destino, fontes para este grupo, a interface através da qual o datagrama deve ser recebido e as interfaces através das quais este deve ser enviado, além de um

campo TTL que representa o número mínimo de saltos pelos quais um datagrama irá passar até atingir um membro de um determinado grupo. A informação deste *cache* de envio é atualizada periodicamente, e mantida enquanto os recursos do sistema estiverem disponíveis (i.e., memória) ou até haver uma mudança de topologia.

Os roteadores MOSPF devem desconsiderar os roteadores unicast na construção de suas árvores de envio. Este fato pode trazer alguns problemas, datagramas multicast podem atravessar um caminho que não seja o ótimo caso existam roteadores unicast no melhor caminho; mesmo que haja conectividade unicast para um destino, pode não ser possível a conexão multicast; o envio de um datagrama unicast e outro multicast entre os mesmos nós fonte e destino pode seguir caminhos diferentes.

O roteamento inter-área envolve o caso onde a fonte de um datagrama e pelo menos um de seus destinos residem em áreas OSPF diferentes. As maiores diferenças são relacionadas com a forma como a informação dos membros dos grupos é propagada e o modo como a árvore de distribuição é construída.

No MOSPF, um sub-conjunto dos roteadores de fronteira das áreas, os *Area Border Routers* (ABRs), funcionam como transmissores de tráfego inter-área. Este roteador é conhecido como Multicast ABR (MABR). Na hierarquia OSPF, existe uma área considerada a área *backbone*, ou área 0, e outras áreas OSPF "comuns" (ou áreas OSPF, simplesmente). Um roteador MABR conecta uma área OSPF à área OSPF *backbone*. Um transmissor multicast inter-área é responsável pelo envio de informação sobre os membros dos grupos entre as áreas e de datagramas multicast.

Para permitir o envio de tráfego multicast entre áreas, os roteadores MABR utilizam o conceito de receptor multicast coringa (*wildcard receiver*). O receptor multicast coringa é um roteador que recebe todo o tráfego multicast em sua área, não importando os membros dos grupos. Para tanto, o receptor coringa injeta em sua área OSPF LSAs com o *flag* "receptor coringa" ligado. Este *flag* indica que "o roteador possui membros conectados para todos os grupos". Em áreas comuns, todos os roteadores multicast inter-área (MABRs) são coringas, injetando LSAs com o *flag* "receptor coringa" ligado na área comum. Isto garante que todo o tráfego multicast gerado numa área comum será enviado para o transmissor multicast inter-área (o MABR estará conectado a qualquer árvore multicast dentro da área), e se necessário então para a área *backbone*.

Para completar o roteamento inter-área, o MOSPF define mais um tipo de mensagem de estado de enlace (LSA), o LSA de Resumo de Grupos (*Summary Membership LSA*). Este LSA lista todos os grupos multicast de interesse dentro de uma área OSPF. Os roteadores multicast inter-área injetam LSAs de Resumo de Grupos na área 0. Desta forma, os roteadores na área *backbone* possuem informação de grupos e membros para todas as áreas. Assim o tráfego multicast pode então ser enviado para membros de todas as áreas.

A Figura 8 ilustra o funcionamento do roteamento MOSPF inter-área. Suponha que a fonte $S1$ na Área 1 começa a enviar dados para o grupo multicast G_B. Ao receber o tráfego enviado a G_B, cada roteador na Área 1 calcula a árvore de caminhos mais curtos (utilizando o algoritmo de Dijkstra) com raiz em $S1$ e cobrindo todos os receptores de G_B. Na Área 2, a fonte $S2$ começa a enviar dados para o grupo G_A. Mais uma vez, os roteadores podem construir uma árvore de distribuição (com raiz $S2$) dentro da Área 2 graças à informação de receptores interessados difundida através dos LSAs de grupo multicast. Esta árvore conecta os Receptores 4 e 5 à fonte $S2$. Até então, os roteadores na Área 2 não sabem que existem receptores interessados no grupo G_A na Área 1.

Neste cenário, os roteadores $MABR1$ e $MABR2$ são configurados como receptores coringa, injetando LSAs com o *flag* correspondente ligado nas Áreas 1 e 2, respectivamente. Desta forma, o roteador $MABR1$ é enxertado na árvore multicast de $S1$, e $MABR2$ é conectado à árvore de $S2$. Além disso, o roteador $MABR1$ injeta na Área 0 LSAs de resumo listando os grupos G_A e G_B. O roteador $MABR2$ injeta LSAs informando que a Área 2 possui receptores interessados no grupo G_A. Os roteadores na Área 0 utilizam a informação contida nestes LSAs de resumo de

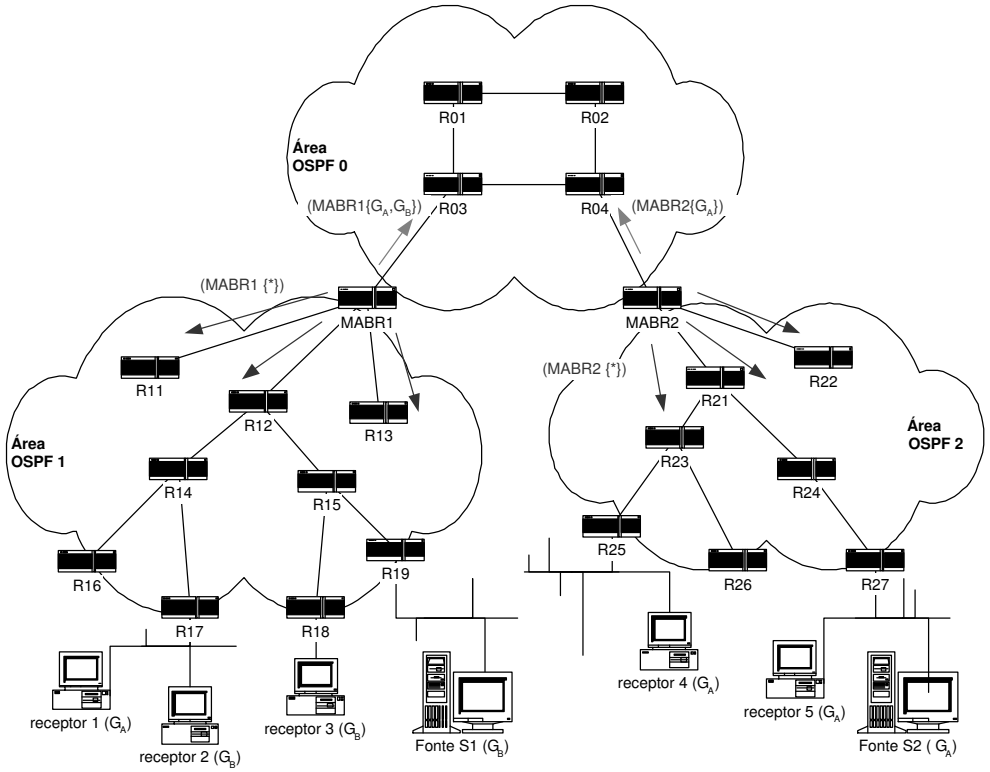


Figura 8: Funcionamento do MOSPF inter-área.

grupo para saber quais roteadores MABR devem ser incluídos na árvore SPT do *backbone*, para quais fontes. Desta forma, o tráfego injetado por $MABR2$ na Área 0 (fonte $S2$, grupo G_A) pode ser encaminhado para o roteador $MABR1$.

No caso de roteamento multicast inter-área, não é realizada a construção da árvore exata de caminhos mais curtos, árvores incompletas são construídas porque a informação detalhada dos membros dos grupos e da topologia da rede não é distribuída através das áreas OSPF. Desta forma, pode haver o transporte de tráfego multicast desnecessário, pois o tráfego é sempre enviado ao roteador MABR como resultado do mecanismo de "receptor coringa".

Quanto ao roteamento inter-domínios (inter-AS, ou entre Sistemas Autônomos), o funcionamento do MOSPF é bastante semelhante ao roteamento inter-área. Os roteadores de *backbone* são informados através dos LSAs de resumo de grupo sobre quais MABRs possuem receptores interessados em quais grupos. Alguns dos roteadores na fronteira entre estes Sistemas Autônomos (ASBRs - *Autonomous System Border Routers*) são designados como transmissores multicast inter-AS (*Multicast AS Border Router* – MASBR). Quando recebe tráfego multicast proveniente de fora do AS, o MASBR re-encaminha este tráfego para os roteadores MABR, de acordo com a necessidade. Os roteadores MASBR também utilizam o mecanismo de "receptor coringa" nos LSAs que propagam dentro da Área 0, desta forma eles recebem todo o tráfego multicast da área e podem anunciá-lo no nível inter-domínio.

Em resumo, o MOSPF é um protocolo para uso dentro de um único domínio, e exige que este domínio execute o protocolo de roteamento unicast OSPF. O MOSPF possui problemas de escalabilidade, por um lado porque a informação de estado do enlace e grupos de interesse inunda a rede periodicamente e por outro lado porque o algoritmo de Dijkstra deve ser executado para todas as fontes multicast – grupos multicast muito dinâmicos podem prejudicar o desempenho do sistema.

2.5 O Protocolo CBT

O protocolo Core Based Trees (CBT) [27, 28] foi o primeiro a utilizar árvores centradas. A árvore de distribuição construída pelo CBT é bi-direcional, e compartilhada por todas as fontes de um mesmo grupo multicast, diminuindo a quantidade de estado armazenada nos roteadores. A raiz da árvore CBT é um nó específico da rede (chamado *core*), em vez da fonte de dados como no DVMRP ou MOSPF. Os roteadores na árvore de distribuição armazenam uma entrada por grupo, em vez de uma entrada por $(fonte, grupo)$ em suas tabelas de roteamento. Desta forma, o CBT é mais escalável que o DVMRP e OSPF, em termos da quantidade de estado armazenada nos roteadores.

Quando um receptor deseja conectar-se a um grupo multicast, ele envia uma mensagem de enxerto (*join*) na direção do *core*. Esta mensagem cria estado em cada roteador atravessado e faz com que uma mensagem de reconhecimento seja enviada ao roteador anterior, construindo a árvore multicast. Quando uma fonte envia dados para o grupo multicast (na direção do *core*), estes são distribuídos na árvore a partir do primeiro roteador a recebê-los. Este roteador reenvia os dados em todas as interfaces de rede pertencentes à árvore multicast, exceto a interface pela qual os dados foram recebidos, de forma a evitar *loops*.

A Figura 9 ilustra o mecanismo de construção da árvore CBT. Os receptores localizados nas redes $N3$ e $N4$ anunciam seu interesse em receber o grupo G através do protocolo IGMP. Os roteadores designados respectivos, $DR3$ e $DR4$, enviam mensagens $join(G)$ em direção ao roteador *Core* associado ao grupo G. As mensagens *join* deixam estado para o grupo G nos roteadores $R4$, $R6$ e $R5$.

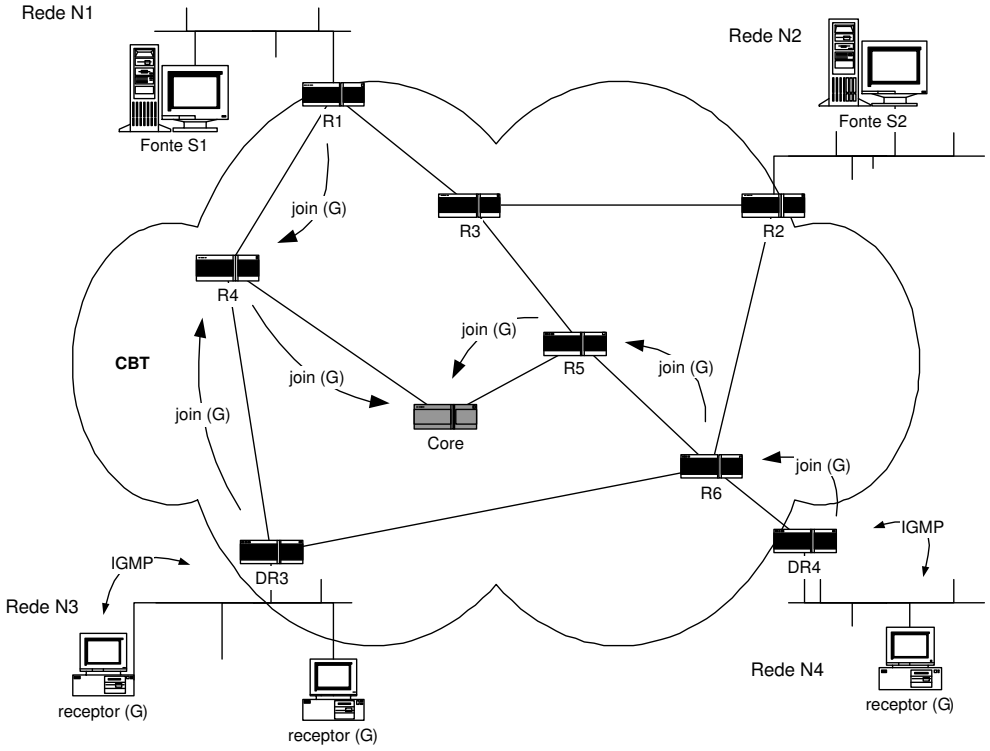


Figura 9: Construção da árvore multicast CBT.

Se um membro do grupo envia dados ao grupo, os pacotes são encaminhados pelo seu roteador local a *todos* os seus vizinhos CBT que estão conectados à árvore de distribuição de G. Este é o caso da fonte $S1$ na Figura 10. Cada roteador que recebe um pacote multicast para o grupo G o reenvia em todas as interfaces de saída que estão na árvore G, exceto aquela por onde o

pacote chegou. Por esta razão, a árvore CBT é compartilhada e bi-direcional, os pacotes podem fluir sobre a árvore dos membros na direção do *Core* ou na direção contrária, despendendo do posicionamento da fonte. A árvore é compartilhada, pois todas as fontes a utilizam – e todos os receptores recebem todas as fontes.

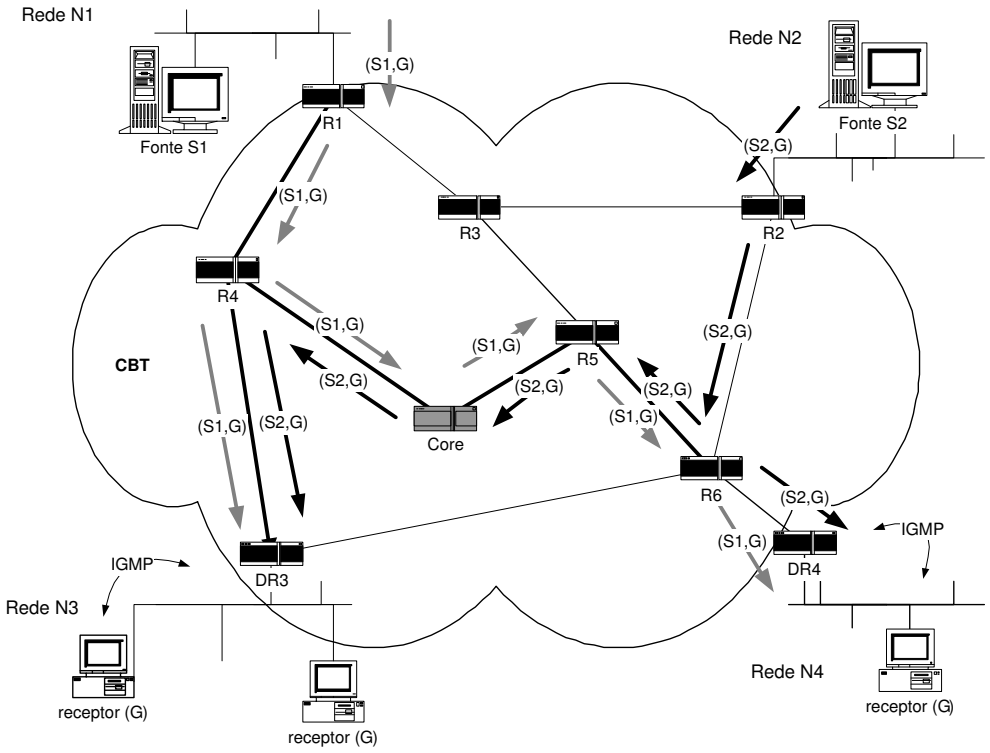


Figura 10: Distribuição de dados na árvore CBT.

O modelo de serviço IP multicast não exige que uma fonte seja membro do grupo multicast, desta forma é possível no CBT que o roteador local da rede de uma fonte para o grupo G não esteja conectado à árvore de G. Por exemplo, a fonte $S2$ na Figura 10 (nesta figura, a linha cheia representa os enlaces que pertencem à árvore G). Neste caso, o pacote é encaminhado salto a salto, na direção do roteador *Core*. Se o pacote, eventualmente, atingir um roteador que esteja na árvore G, ele é encaminhado na árvore a partir deste ponto. No caso da fonte $S2$, o tráfego multicast é distribuído sobre a árvore a partir do roteador $R6$ (Figura 10).

O protocolo CBT permite que vários roteadores *core* sejam especificados, criando redundância para o caso em que o *core* não esteja disponível. O CBT é eficiente em termos do estado armazenado nos roteadores. Apenas os roteadores que pertencem à árvore de distribuição para o grupo G mantêm estado para G (diferentemente do DVMRP e MOSPF), e nenhum roteador mantém estado por fonte. Assim, o CBT é mais escalável que protocolos de inundação-e-poda, especialmente para grupos esparsos – onde apenas algumas partes da rede possuem membros.

Apesar de suas vantagens, o CBT possui algumas limitações. O algoritmo pode resultar na concentração de tráfego próximo aos roteadores centrais (raízes) das árvores, uma vez que o tráfego de todas as fontes atravessa os mesmos enlaces à medida em que se aproximam do centro. Além disso, uma única árvore de envio compartilhada pode criar rotas sub-ótimas para determinados fluxos, resultando em maiores atrasos, uma questão crítica para aplicações multimídias. Conseqüentemente, no CBT a localização do *core* é crítica pois define a forma da árvore de distribuição. Além disso, o CBT não define um mecanismo de mapeamento entre grupos multicast e roteadores *core*.

2.6 O Protocolo PIM

O PIM (*Protocol Independent Multicast*) [29] é constituído de dois protocolos diferentes, de modo denso (PIM-DM) e modo esparso (PIM-SM). O PIM-DM é destinado a redes onde os membros do grupo estão densamente distribuídos, sendo funcionalmente muito similar ao DVMRP. Desta forma, o PIM-DM utiliza a inundação periódica da rede, com mensagens de poda. Já o PIM-SM foi projetado para o roteamento em grande escala, e assume que a distribuição de receptores na rede é esparsa. Neste caso, protocolos que utilizam a inundação da rede são inadequados. O PIM-SM constrói árvores compartilhadas baseado em mensagens *join*, como o CBT, porém as árvores PIM-SM são uni-direcionais. A fonte encapsula os dados em unicast e os envia a um nó de *rendez-vous* (RP) que em seguida os distribui na árvore multicast. É possível no PIM-SM mudar para uma árvore por fonte, para fontes com alta taxa de transmissão. Desta forma, a localização do RP é menos crítica no PIM-SM que o *core* no CBT.

2.6.1 O Protocolo PIM *Dense Mode*

Enquanto a arquitetura do PIM foi dirigida para a necessidade de prover árvores de distribuição de modo-esparso escaláveis, esta define também um novo protocolo de modo-denso (PIM-DM). Este deve vir a ser utilizado em ambientes onde a faixa de transmissão disponível seja abundante e os grupos densos, como numa LAN de um campus universitário.

O PIM-DM é similar ao DVMRP do ponto de vista do algoritmo utilizado, o RPM. No entanto, há algumas diferenças importantes. O PIM-DM se apóia na existência de um protocolo de roteamento unicast para se adaptar a mudanças de topologia, mas é independente dos mecanismos do protocolo específico. Por outro lado, o DVMRP contém um protocolo de roteamento integrado que utiliza seu próprio mecanismo de troca de mensagens para atualização das tabelas de roteamento. O MOSPF usa a informação contida na base de dados de estado dos enlaces do OSPF, sendo o MOSPF específico apenas para o OSPF. Diferentemente do DVMRP, que calcula um conjunto de interfaces filhas para cada par $(fonte, grupo)$, o PIM-DM simplesmente envia o tráfego multicast em todas as interfaces no sentido descendente do fluxo até que mensagens de poda explícita sejam recebidas. O PIM-DM convive com a duplicação de pacotes para eliminar a dependência do protocolo de roteamento e evitar o *overhead* associado com a construção de uma base de dados contendo os enlaces pais/filhos.

A Figura 11 ilustra o funcionamento do processo de inundação e poda implementado pelo PIM-DM. Neste exemplo, o tráfego multicast gerado pela fonte $S1$, localizada na rede $N1$, inunda toda a rede. Cada roteador da rede, ao receber o tráfego, checa se este chegou pela interface utilizada pelo roteador para chegar à rede $N1$ (teste RPF). Em caso afirmativo, o roteador reenvia o pacote em *todas* as suas interfaces de saída (onde estejam conectados vizinhos PIM-DM). Desta forma, todos os enlaces da rede são utilizados, diferentemente do DVMRP com sua árvore TBT. Além disso, alguns pacotes podem ser transportados em um enlace nos dois sentidos, por exemplo entre os roteadores $R4$ e $R5$. Este é o preço pago pelo PIM por não implementar um protocolo de roteamento unicast, como o DVMRP.

Após a inundação inicial, alguns roteadores da rede enviarão mensagens de poda (*prune*) para parar o tráfego multicast enviado a $G1$. Este é o caso de roteadores que não possuem receptores interessados no tráfego multicast de $G1$, como $R7$. Roteadores que não possuem vizinhos interessados em $G1$ (ou seja, nós que não possuem filhos na árvore) também enviam mensagens de poda (este é o caso de $R8$ e $R3$). Além disso, roteadores que receberam o tráfego de $S1$ através de uma interface diferente da interface correta pelo teste RPF também enviam podas nestas interfaces, $R4$, $R5$ e $R9$ são exemplos deste caso.

O resultado do processo de inundação e poda do PIM-DM é uma árvore SPT reversa, ou seja, resultado da união dos caminhos mais curtos dos receptores para a fonte. Embora o tráfego de dados para $G1$ não esteja chegando a todos os roteadores da rede, existe estado para esta

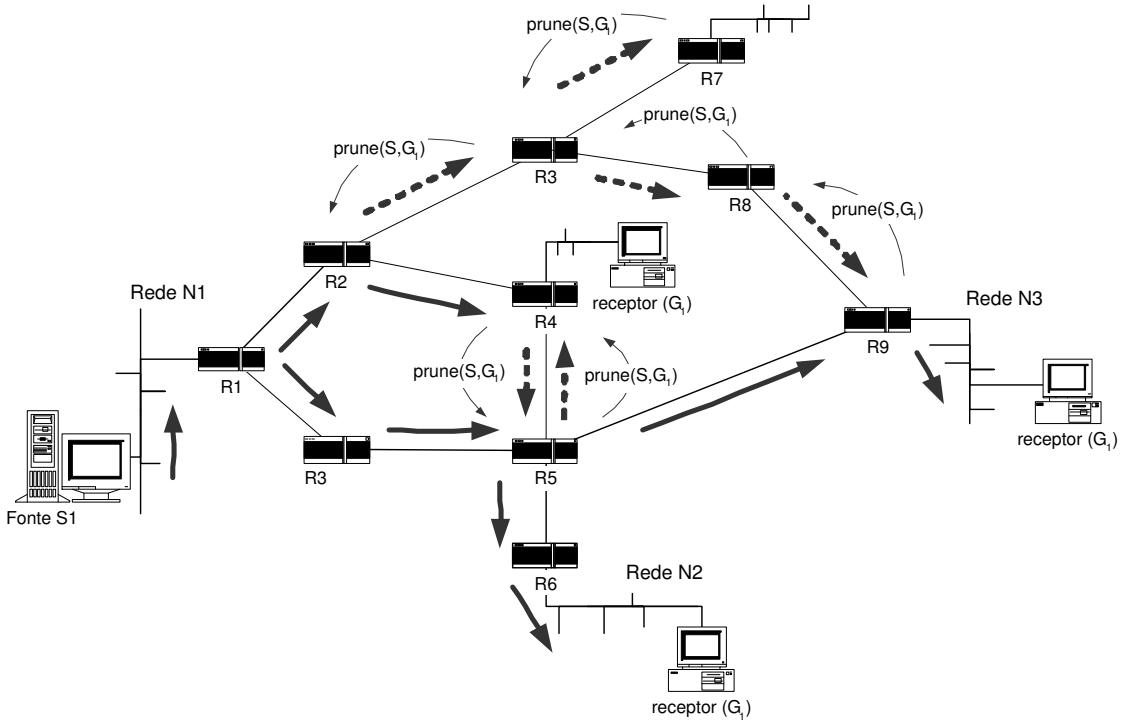


Figura 11: Distribuição de dados no protocolo PIM-DM.

fonte/grupo $(S1, G1)$ em *todos* os roteadores da rede. O estado $(S1, G1)$ continuará armazenado enquanto a fonte $S1$ estiver transmitindo.

No PIM-DM, as podas (estados *prune*) são estados voláteis, que expiram a cada 3 minutos. Isto ocasiona a inundação de toda a rede, exatamente como no caso inicial, com esta mesma freqüência. Por este motivo, o PIM-DM é adequado apenas para redes onde os receptores estão *densamente* distribuídos.

2.6.2 O Protocolo PIM *Sparse Mode*

O PIM *Sparse Mode* (PIM-SM) [30] foi desenvolvido para situações onde várias estações desejam participar de uma conferência multicast mas a inundação periódica de toda a rede com tráfego multicast não se justifica, seja porque o número de receptores é pequeno, ou pelos receptores estarem concentrados em certas regiões da topologia. Para tratar este tipo de cenário, de distribuição esparsa dos receptores, o PIM-SM foi projetado para limitar o tráfego multicast de forma que somente os roteadores interessados em receber tráfego de um determinado grupo o enxerguem.

O PIM-SM é diferente dos algoritmos multicast de modo-denso em dois aspectos principais. Roteadores com membros diretamente conectados ou membros no caminho descendente do fluxo se conectam à árvore de distribuição através da transmissão de mensagens *join* explícitas. Se um roteador não se tornar parte de uma árvore de distribuição pré-definida (a árvore compartilhada), não receberá o tráfego multicast endereçado para o grupo. Em contraste, os algoritmos de modo-denso assumem a pertinência do grupo no sentido descendente do fluxo e continuam a enviar o tráfego multicast até que mensagens explícitas de poda sejam recebidas. O procedimento padrão adotado nos algoritmos de modo-denso é o envio do tráfego, enquanto que no modo-esparso o padrão é bloquear o tráfego a menos que este seja explicitamente requisitado. Além disso, as árvores construídas são árvores compartilhadas em vez de árvores por fonte.

O PIM-SM emprega, como o CBT, nós centrais à rede que são utilizados como raiz da árvore

compartilhada. No PIM, estes roteadores especiais são chamados de pontos de *rendez-vous* (RP), este é o lugar da rede onde os receptores "encontram" as fontes. Cada grupo multicast (endereço Classe D) é associado a um RP da rede, através de um mecanismo de configuração. Para cada grupo, existe apenas um RP ativo. Cada receptor que deseja se conectar a um grupo contata o roteador ao qual está diretamente conectado, através do protocolo IGMP. Este roteador (DR) por sua vez entra na árvore de distribuição do grupo multicast através do envio de uma mensagem *join* para o RP primário do grupo. As mensagens *join* enviadas na direção do RP constroem, de forma explícita, uma árvore de distribuição.

No PIM-SM, independentemente do número e da localização dos receptores, as fontes se "registram" com o RP e enviam uma única cópia de cada pacote multicast para os receptores conectados ao RP. Da mesma forma, independentemente do número de fontes, os receptores devem sempre se conectar ao RP para receber o tráfego multicast enviado ao grupo. O funcionamento do PIM-SM é baseado nesta árvore compartilhada com raiz no roteador RP.

Este modelo requer que os roteadores mantenham algum estado (a lista dos RPs, contendo o mapeamento endereço multicast - RP) antes da chegada dos pacotes multicast. Por outro lado, protocolos multicast de modo-denso são orientados pelos dados, uma vez que eles não definem qualquer estado para o grupo até a primeira chegada de pacotes.

A especificação do PIM-SM permite, por outro lado, a troca da árvore compartilhada ($(*, G)$) por uma árvore por fonte. Esta troca de árvore pode ser configurada, tomando geralmente como parâmetro a taxa de envio de dados da fonte. A partir de um certo valor limite da taxa de dados recebidos de uma determinada fonte, os roteadores de último salto (roteadores diretamente conectados a receptores) podem tomar a decisão de trocar de árvore. A partir de então, o roteador passa a emitir mensagens PIM $join(S, G)$ na direção da fonte S. Estas mensagens trafegam salto por salto até o roteador de primeiro salto conectado à fonte S. O roteador de último salto não pára, no entanto, de enviar mensagens $join(*, G)$ para o RP, pois isso faria com que tráfego de outras fontes ativas para o grupo G (e fontes que venham a ficar ativas) não fosse recebido. Por outro lado, o roteador de último salto deve enviar uma mensagem de poda especial ao RP, $RP-bit-prune(S, G)$, para evitar que o tráfego gerado por S seja recebido em duplicata.

As Figuras 12 e 13 ilustram o funcionamento do protocolo PIM-SM. Suponha que existem duas fontes de tráfego, $S1$ e $S2$, localizadas nas redes $N1$ e $N2$, respectivamente. Ambas enviam dados para o grupo multicast G, que possui receptores nas redes $N3$ e $N4$. A partir do momento em que o roteador de primeiro salto de $S1$, $R1$, detecta uma fonte de dados ativa para o grupo G, este "registra" a fonte $S1$ junto ao RP (através do envio de uma mensagem PIM $register(S1, G)$ ao RP). Na realidade, o roteador $R1$ encapsula os dados enviados por $S1$ dentro das mensagens PIM *register*, mensagens que são enviadas em unicast ao RP. Quando o RP recebe as mensagens $register(S1, G)$, ele desencapsula os dados multicast e os envia sobre a árvore compartilhada. O mesmo processo é usado para qualquer fonte que começa a enviar dados endereçados a G, como $S2$.

A construção dos ramos da árvore compartilhada é realizada pelo envio de mensagens $join(*, G)$ enviadas pelos roteadores de último salto, na direção do RP. Por exemplo, o roteador $DR3$ detecta a presença de receptores interessados no grupo G através do protocolo IGMP. $DR3$ envia então um $join(*, G)$ ao RP. O mesmo ocorre para o roteador $DR4$, construindo a árvore compartilhada mostrada na Figura 12.

Suponha que os roteadores $DR3$ e $DR4$ foram configurados com um valor $x \neq 0$ para a taxa de transmissão da fonte, a partir da qual ocorrerá a troca para árvore por fonte. Suponha também que a taxa de transmissão de $S1$ é maior que x, e que a taxa de $S2$ é menor que x. Neste caso, o roteador $DR3$ toma a decisão de trocar da árvore compartilhada para a árvore por fonte, enviando mensagens $join(S1, G)$ à fonte $S1$. Esta mensagem cria estado $(S1, G)$ nos roteadores $R4$ e $R1$. Além disso, $DR3$ envia mensagens de poda, $RP-bit-prune(S1, G)$ ao roteador RP. Isto evita que o tráfego de $S1$ seja recebido em duplicata na rede $N3$. O roteador

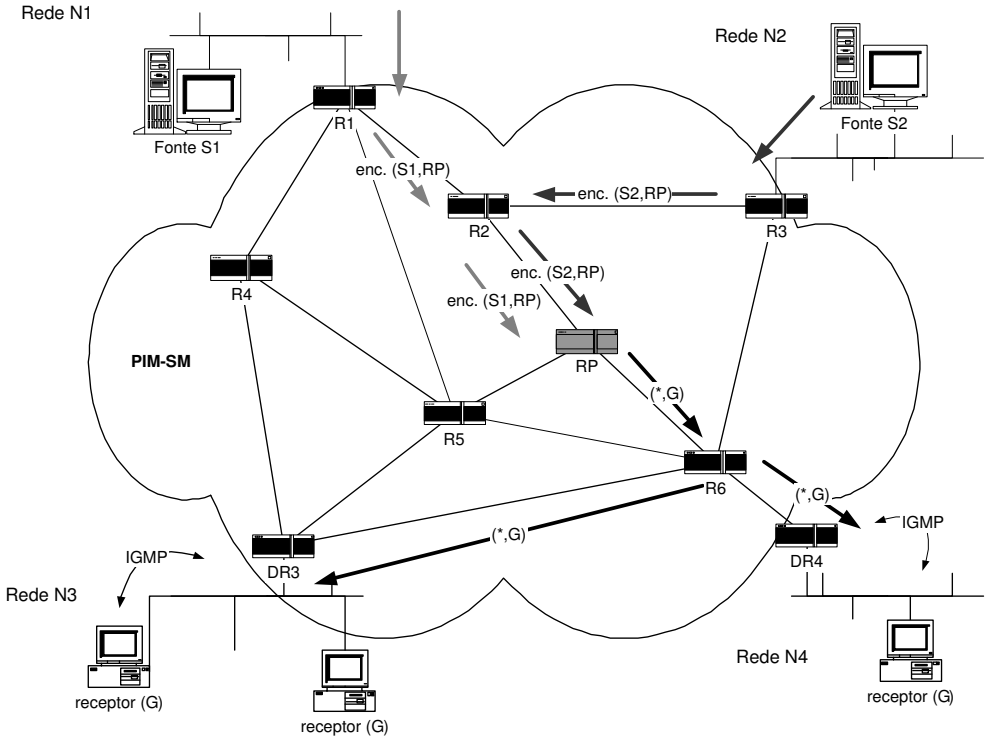


Figura 12: Árvore compartilhada no PIM-SM.

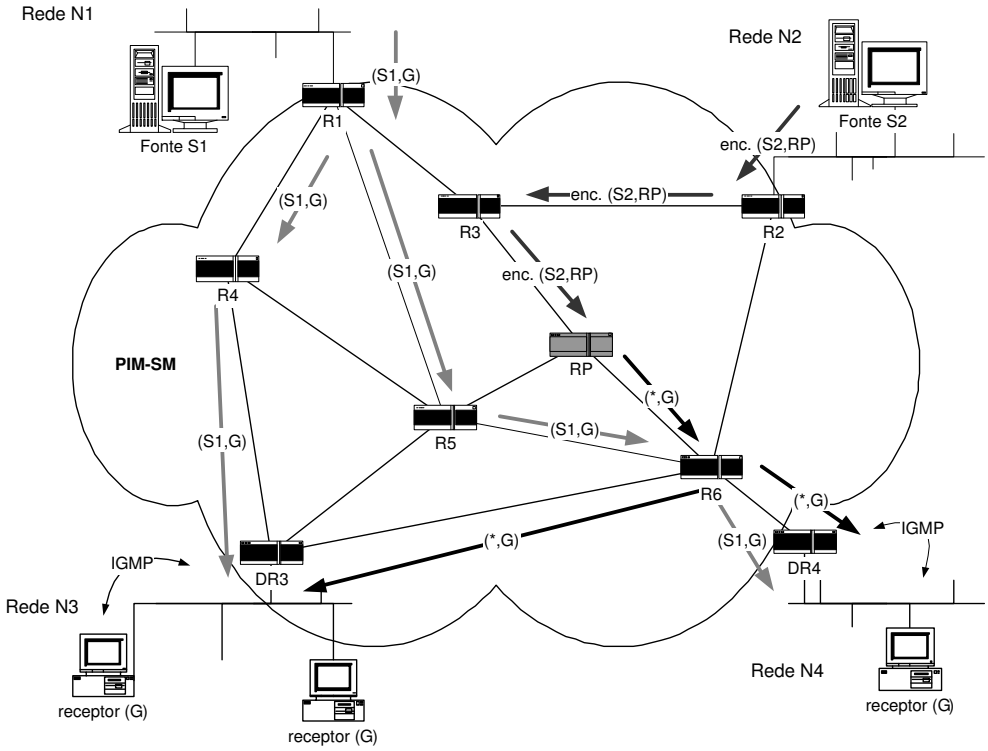


Figura 13: Árvore por fonte no PIM-SM.

$DR4$ toma a mesma decisão, criando estado $(S1, G)$ nos roteadores $R6$ e $R5$. A partir de então, o tráfego endereçado a G enviado por $S1$ segue utilizando uma árvore de distribuição SPT reversa, enquanto o tráfego de $S2$ continua a ser transportado na árvore compartilhada.

Além da troca entre árvore compartilhada e por fonte, disparada pelos roteadores de último salto, é possível também que o RP se conecte ao roteador de primeiro salto através de uma árvore por fonte. De forma semelhante, configura-se um valor para a taxa de transmissão da fonte a partir do qual o RP irá enviar uma mensagem $join(S, G)$ à fonte. Assim, é criado mais um ramo da árvore reversa SPT entre a fonte e o RP. Desta forma, é possível a criação de uma árvore SPT completa, mesmo que essa atravesse o RP, que armazenará estado (S, G) neste caso.

A idéia por trás da utilização de árvores por fonte, apenas para determinadas fontes, é motivada pelo fato de árvores compartilhadas poderem apresentar atraso maior que árvores por fonte (pois os receptores têm que "passar" pelo RP). Para minimizar os problemas causados por uma má localização do RP dentro da topologia da rede, criou-se a possibilidade de troca da árvore para uma árvore por fonte.

3 Roteamento Multicast Inter-domínio

A implementação do IP Multicast no nível inter-domínio é complexa [31]. Uma vez que o PIM-SM se baseia no roteamento unicast para construir a árvore multicast (assumindo que o caminho *reverso* unicast é adequado para o tráfego multicast), mensagens de controle da árvore podem ser recebidas por roteadores não-multicast, dificultando a operação do protocolo. Além disso, a utilização do PIM-SM no inter-domínio tem ainda dois problemas: o projeto de um mecanismo escalável de mapeamento de grupos multicast para RPs e a inter-dependência introduzida pela própria utilização de RPs entre os ISPs (*Internet Service Provider*). A localização de um RP em um ISP alheio pode ser inaceitável, de um ponto de vista estratégico.

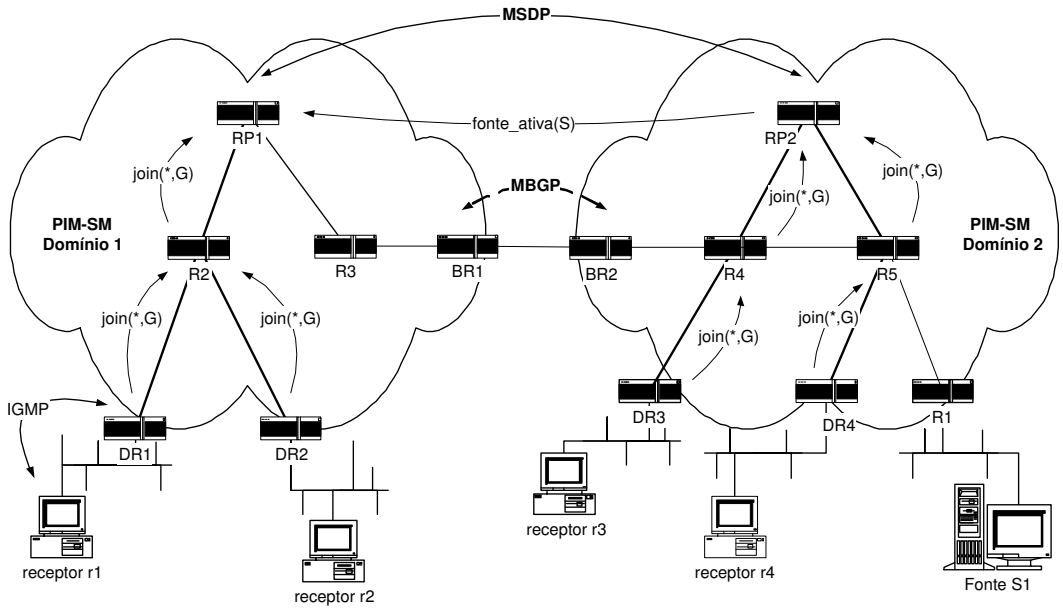
3.1 A Arquitetura MBGP/MSDP

A solução de curto-prazo para os problemas acima é a combinação dos protocolos MBGP (*Multiprotocol Extensions for BGP-4*) [32] e MSDP (*Multicast Source Discovery Protocol*) [33]. O MBGP permite que múltiplas tabelas de roteamento possam ser mantidas para diferentes protocolos. Desta forma, os roteadores podem construir tabelas diferentes para rotas unicast e multicast. O MBGP armazena as rotas multicast em uma tabela chamada M-RIB (*Multicast Route Information Base*).

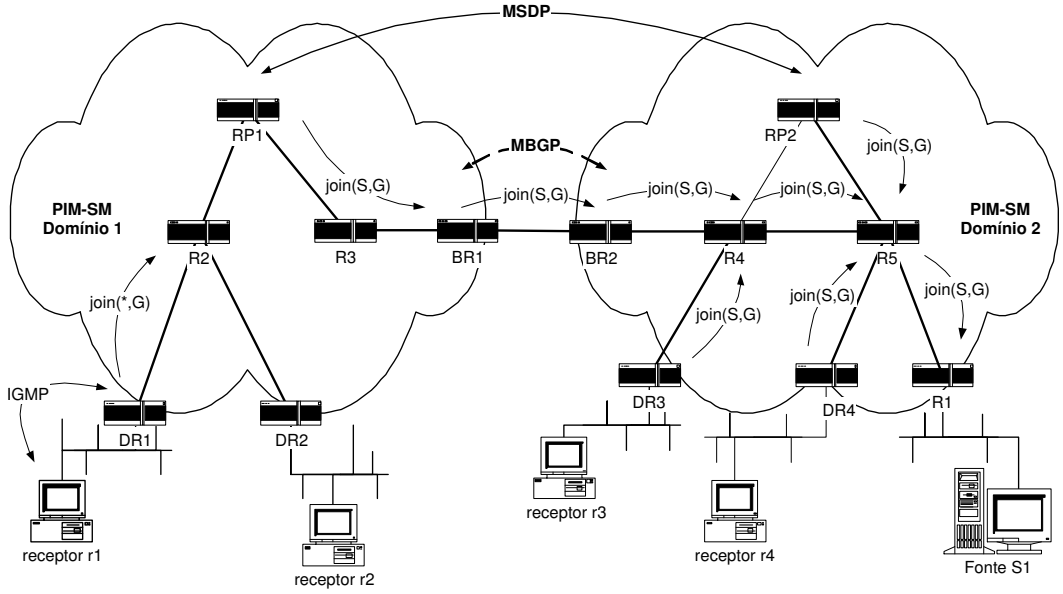
O protocolo MSDP tenta solucionar o problema de inter-dependência entre os ISPs. Cada ISP roda o protocolo PIM-SM dentro de seu domínio, com um conjunto próprio de RPs para todos os grupos dentro de seu domínio. Os roteadores RPs de um mesmo domínio são interconectados, assim como conectados a RPs de outros domínios, utilizando conexões de controle MSDP, formando uma malha.

Quando uma fonte começa a emitir, o RP em seu domínio notifica RPs em outros domínios desta fonte ativa. Receptores em outros domínios enviam mensagens *join* específicas para esta fonte.

A Figura 14 mostra o processo de construção da árvore multicast inter-domínio. Suponha um grupo multicast G, com uma fonte ativa, $S1$, localizada no Domínio 2 e receptores distribuídos nos dois domínios. Dentro do Domínio 1, os receptores $r1$ e $r2$ se conectam ao roteador $RP1$, ponto de *rendez-vous* do Domínio1, através do envio de mensagens $join(*, G)$. Processo semelhante ocorre dentro do Domínio 2 para os receptores $r3$ e $r4$. Duas árvores compartilhadas, isoladas, são construídas (na Figura 14(a), as linhas cheias indicam os enlaces pertencentes às árvores). Quando a fonte $S1$ começa a emitir dados, seus pacotes são encapsulados dentro de mensagens $PIM - register$ e enviados a $RP2$, de acordo com o comportamento normal do PIM-SM. O roteador $RP2$ desencapsula os dados e os envia, na árvore compartilhada, aos receptores $r3$ e $r4$.



(a) Construção das árvores PIM-SM intra-domínio.



(b) Construção da árvore inter-domínio PIM-SM/MSDP.

Figura 14: Funcionamento do PIM-SM/MSDP/MBGP.

Além disso, o roteador $RP2$ anuncia a existência de uma fonte ativa a todos os RPs conectados à malha MSDP. O roteador $RP1$ recebe então uma mensagem dizendo que a fonte $S1$ está ativa para o grupo G (Figura 14(a)). Ao receber a notificação, roteadores que possuem receptores interessados no grupo G, como é o caso de $RP1$, enviam uma mensagem *join* específica para a fonte $S1$ ($join(S1, G)$), através dos roteadores de inter-domínio, que rodam o protocolo MBGP (Figura 14(b)). Esta mensagem constrói um ramo da árvore que liga $R1$, roteador de primeiro salto de $S1$, a $RP1$. Desta forma, o tráfego será transmitido entre os domínios, para $RP1$, e chegará aos receptores $r1$ e $r2$ (Figura 15). Além disso, supondo que a fonte $S1$ ultrapassa o limite de taxa de transmissão configurado para o PIM-SM, os roteadores de último salto DR3 e DR4, emitem mensagens $join(S1, G)$, trocando a árvore compartilhada por uma árvore específica à fonte $S1$. Neste exemplo, $RP2$ toma a mesma decisão e constrói mais um ramo da árvore SPT (Figura 14(b)). O envio de dados através das árvores específicas ocorre como mostrado na Figura 15.

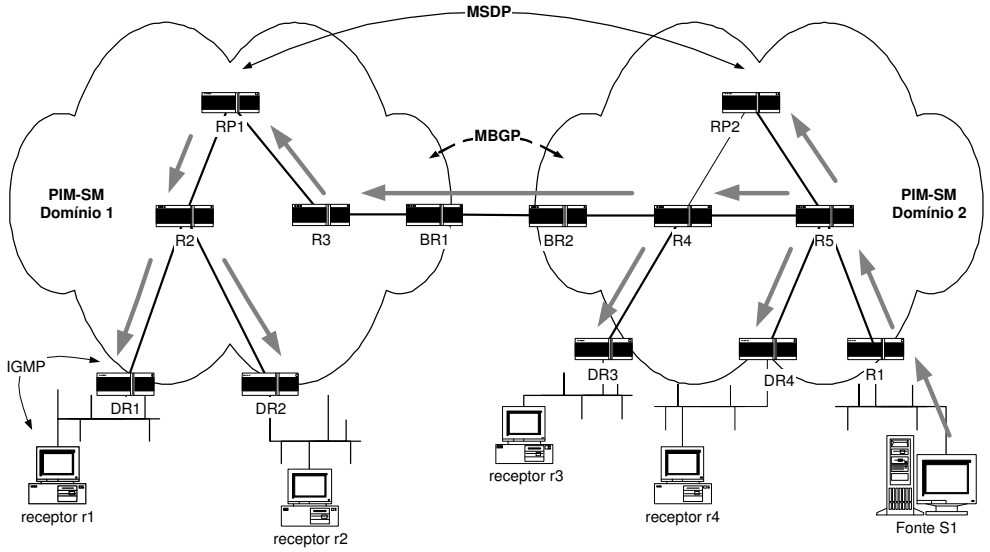


Figura 15: Envio de dados através do PIM-SM/MSDP.

O protocolo MSDP utiliza as mensagens *join* específicas normais do PIM-SM, passando entre os domínios através do protocolo MBGP e se conectando à fonte de dados. Por outro lado, o MSDP só constrói árvores compartilhadas dentro de cada domínio, (nunca no inter-domínio) evitando que se dependa de RPs remotos para o envio de tráfego entre dois membros do mesmo domínio.

Embora o MSDP resolva o problema da interconexão de domínios PIM-SM, ele possui algumas desvantagens. Um problema de escalabilidade é que os RPs em *todos* os domínios devem ser notificados de *toda* fonte que comece a enviar dados. Além disso, alguns RPs irão guardar essa informação em um *cache* de forma a que membros que se interessem pelo grupo após o início da transmissão possam fazer com que seus RPs correspondentes enviem mensagens *join* específicas à fonte. Considerando que o MSDP é um protocolo inter-domínio, esta característica compromete sua escalabilidade com relação ao número de fontes e grupos ativos. Além disso, para garantir que os primeiros pacotes emitidos pela fonte sejam recebidos pelos membros em outros domínios, os dados devem ser encapsulados e enviados com as mensagens de "fonte-ativa" do MSDP, a *todos* os RPs conectados à malha MSDP. Além disso, se os dados não forem encapsulados, fontes que enviam dados em rajadas, espaçadas de alguns minutos, poderiam nunca ser recebidas por membros em outros domínios, pois o estado de roteamento $(S1, G)$ poderia ter expirado a cada vez que a enviasse.

Portanto, o MSDP não é uma solução de longo prazo, mas foi implementado por alguns ISPs, pois resolve os problemas de inter-conexão de domínios PIM-SM. Por outro lado, a partir do momento em que o número de usuários multicast se tornar mais expressivo, a utilização do MSDP será inviável. O grupo de trabalho do IETF responsável pela padronização do MSDP, após várias revisões do padrão, decidiu não mais trabalhar na especificação do protocolo.

3.2 O Protocolo BGMP

O *Border Gateway Multicast Protocol* (BGMP) [14] é um protocolo de roteamento multicast inter-domínio que tenta resolver alguns dos problemas das outras soluções existentes. O BGMP reúne algumas das idéias dos protocolos de roteamento anteriores, e tenta ser mais amigável do ponto de vista do provedor de serviços – seu projeto se assemelha ao protocolo BGP [13], o único protocolo de roteamento unicast inter-domínio utilizado atualmente na Internet. O BGMP constrói árvores compartilhadas para os grupos multicast ativos e permite que os domínios receptores construam ramos específicos para a fonte no inter-domínio, quando necessário. O comportamento padrão é construir ramos compartilhados no inter-domínio pois em geral a conectividade neste nível é menor que no intra-domínio. Além disso, as árvores são bi-direcionais para diminuir a dependência de terceiros. O BGMP está atualmente em curso de padronização pelo IETF [34].

O BGMP é um protocolo inter-domínio que adota alguns princípios de projeto do BGP, familiares dos provedores de serviço. Como o BGP, o BGMP utiliza o protocolo de transporte confiável TCP para transmitir a informação de roteamento. Além disso, a máquina de estados com notificação de erros do BGMP é similar à do BGP.

O BGMP constrói árvores compartilhadas bi-direcionais. Este tipo de árvore permite uma menor quantidade de estados de reenvio multicast nos roteadores, sendo mais escalável que os outros tipos de árvore (por fonte e compartilhadas uni-direcionais). No entanto, para manter compatibilidade com outros protocolos, o BGMP pode construir ramos uni-direcionais específicos por fonte. Árvores bi-direcionais são adequadas para aplicações de tipo NxM (N fontes e M receptores, ou "vários para vários", como por exemplo jogos distribuídos ou vídeoconferências). Árvores uni-direcionais são úteis para aplicações de fonte única e sensíveis ao atraso.

Uma das caraterísticas originais do BGMP é a raíz da árvore compartilhada ser um Sistema Autônomo (AS), não um roteador como no PIM-SM ou CBT. Cada AS deve ser associado a um endereço de grupo multicast. Utilizar o AS ao qual está associado um endereço de grupo específico como raiz da árvore de distribuição deste grupo é razoável, porque existem grandes chances de que este AS possua membros interessados neste grupo. Além disso, utilizar um AS como raiz da árvore, em vez de um roteador, proporciona maior estabilidade e tolerância a falhas. Um número de AS é menos sujeito a mudanças que o endereço de um roteador, além disso, utilizar o número de AS facilita a implementação de redundância dentro do AS.

Dado um endereço multicast, a árvore compartilhada do BGMP tem por raiz o domínio para o qual foi alocada a faixa de endereços que o inclui. Desta forma, o BGMP supõe um mecanismo para descobrir qual o AS responsável por cada endereço multicast. Isto pode ser conseguido através do protocolo MASC [12], como mostrado na Seção 1.2.5, ou através da alocação global de endereços multicast, como é realizado, por exemplo, na arquitetura de endereçamento do IPv6 (Seção 5). O protocolo MASC realiza a alocação temporária de endereços da faixa Classe D do IPv4 e então distribui estas associações através do protocolo *Multiprotocol* BGP (MBGP) [32]. O MBGP permite saber que endereço multicast está associado a qual número de AS, e desta forma saber onde as mensagens *join* devem ser enviadas. Além disso, o MBGP permite que as topologias unicast e multicast sejam diferentes.

A Figura 16 mostra um exemplo de árvore de distribuição construída pelo BGMP. Os roteadores de borda implementam dois protocolos de roteamento multicast, o BGMP no nível inter-domínio, e um protocolo de roteamento intra-domínio, como PIM-SM ou DVMRP, nomeado genericamente como MIGP (*Multicast Interior Gateway Protocol*).

Suponha um grupo multicast, G, criado por uma estação pertencente ao domínio B. A estação obtém para G o endereço multicast `224.0.128.1`, pertencente à faixa de endereços associada ao domínio B. Quando uma estação no domínio C se conecta ao grupo, uma mensagem *join* é enviada pelo MIGP ao componente BGMP do melhor roteador de saída (segundo o BGP) para o endereço `224.0.128.1`, $C1$ na Figura 16.

O roteador $C1$ consulta sua tabela de rotas de grupo, G-RIB, e encontra como melhor rota (`224.0.0.0/16`, $A2$) para o destino `224.0.128.1`. $C1$ cria uma "lista de alvos" em sua tabela de encaminhamento multicast com um "alvo pai" e uma lista de "alvos filhos" para o endereço G. O alvo pai é o próximo salto na direção do domínio raiz do grupo G. Um alvo filho identifica um roteador BGMP ou o componente MIGP do qual uma mensagem *join* foi recebida. No caso de $C1$, o alvo pai é $A2$, e o único alvo filho é seu próprio componente MIGP. Esta entrada de encaminhamento, de tipo $(*, G)$, significa que um pacote multicast recebido para G é encaminhado para todos os alvos da lista, exceto aquele pelo qual o pacote foi recebido, implementando desta forma a árvore bi-direcional compartilhada.

Os roteadores de borda BGMP possuem conexões de controle TCP uns com os outros, que são utilizadas para encaminhar mensagens de enxerto (*join*) e poda (*prune*). Após a criação de sua entrada na tabela de encaminhamento, $C1$ envia uma mensagem $join(*, G)$ ao alvo pai $A2$. Ao receber esta mensagem, $A2$ consulta sua tabela de roteamento e encontra a rota (`224.0.128.0/24`, $A3$), indicando $A3$ como próximo salto para a raiz de G. $A2$ cria então uma entrada de encaminhamento, com seu componente MIGP, com $A3$ como alvo pai e $C1$ como alvo filho. Uma vez que $A3$ é um parceiro BGMP interno (roteador BGMP dentro do mesmo domínio), $A2$ transmite o *join* para seu componente MIGP, que toma as medidas necessárias para implementar o encaminhamento de pacotes endereçados a G dentro do domínio A. De maneira semelhante, a mensagem $join(*, G)$ se propaga até o domínio B. Os ramos de árvore representados por flechas bi-direcionais na Figura 16 são construídos, supondo que existem receptores para o grupo G nos domínios D, F e H.

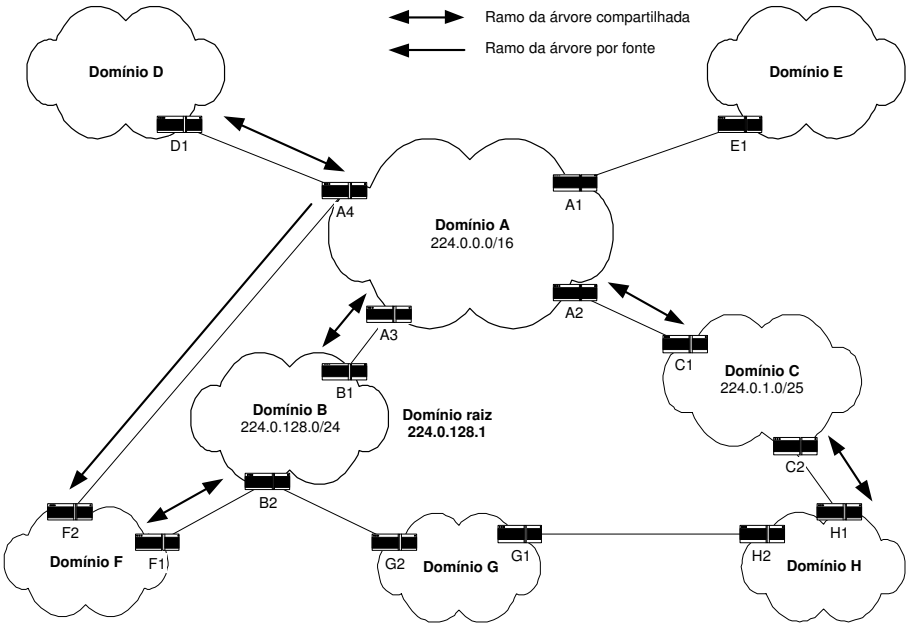


Figura 16: Árvore de distribuição multicast BGMP.

Os dados podem chegar à árvore de G, mesmo quando emitidos por estações que não tenham se conectado a G, respeitando o modelo de serviço IP multicast. Suponha um emissor no domínio E, que não possui membros do grupo G. A estação envia dados endereçados a `224.0.128.1`,

estes dados chegam ao melhor roteador de saída, no caso $E1$, graças ao protocolo de roteamento MIGP. Uma vez que $E1$ não possui estado para este grupo, ele simplesmente reenvia os dados para o próximo salto na direção do domínio raiz de G, o domínio B, para o qual $A1$ é próximo salto neste caso. Uma vez que $A1$ também não possui estado para G, o pacote é reenviado utilizando o MIGP implementado por A de forma a atingir o próximo salto na direção de B. Por exemplo, se o protocolo MIGP for o DVMRP, o pacote multicast será inundado na rede do domínio A, atingindo outros roteadores de borda BGMP. No caso, os roteadores de borda $A2$, $A3$ e $A4$ pertencem à árvore de distribuição de G, desta forma os dados são reenviados na árvore a todos os membros do grupo.

Quando um roteador BGMP, ou seu componente MIGP, não possui conexões com mais nenhum membro do grupo, ele envia uma mensagem de poda (*prune*) para o alvo pai armazenado em sua tabela de encaminhamento. Quando sua lista de alvos filhos se esvazia, o roteador BGMP também procede à poda. Desta forma, o tamanho da árvore de distribuição BGMP é adaptado ao conjunto de membros atualmente conectados ao grupo, de forma dinâmica.

A especificação do BGMP também permite que ramos da árvore específicos a uma fonte sejam construídos. Por exemplo, na Figura 16, o domínio F possui uma conexão ao domínio D que não passa pelo domínio raiz de G, B. Neste caso, supondo que exista uma fonte $S1$, localizada no domínio D, emitindo para o grupo G, o roteador $F2$ pode enviar uma mensagem de enxerto específica, $join(S, G)$, ao roteador $A4$.

A utilização de ramos específicos a uma fonte pode ser útil em casos onde o caminho mais curto para a fonte S a partir de um domínio não coincide com a árvore bi-direcional a partir deste domínio, e o domínio utiliza um protocolo de roteamento como o DVMRP ou PIM-DM, que constrói árvores por fonte dentro do domínio. Neste caso, o teste RPF para a fonte S falharia, já que o tráfego não estaria chegando pelo caminho mais curto na direção de S. A única solução para este problema, sem os ramos específicos, seria o roteador BGMP de entrada do domínio (pela árvore bi-direcional) encapsular os dados e os enviar ao roteador *de saída* para a fonte S (que do ponto de vista do teste RPF, é o roteador correto para recepção do tráfego multicast de S). Com a utilização de ramos específicos, a encapsulação pode ser evitada.

Se endereços multicast alocáveis globalmente existem, então o BGMP pode utilizar qualquer arquitetura de alocação de endereços estática para obter a identificação de um AS a partir de um endereço multicast. A combinanção o BGMP com um espaço de endereçamento suficientemente grande, como o do IPv6, tem o potencial de prover escalabilidade para uma maior gama de aplicações que as soluções de roteamento atuais.

4 Soluções Alternativas e Novos Protocolos

Devido à lenta implantação do serviço multicast na Internet, diferentes soluções alternativas foram propostas [35]. Este capítulo apresenta as principais propostas neste sentido.

4.1 Novos Protocolos de Roteamento

Diferentes propostas foram feitas para simplificar o serviço multicast. O protocolo EXPRESS reduziu a conversação multicast para o caso mais simples de *1 para N*, introduzindo a noção de canal. Esta definição simplifica a alocação de endereços e distribuição de dados, e cobre a maioria das aplicações atuais. O serviço SSM (*source-specific multicast*) foi inspirado no protocolo EXPRESS e está atualmente em fase final de padronização pelo IETF (*Internet Engineering Task Force*). O SSM é implementado pelos protocolos IGMPv3, que suporta a filtragem de fontes, e por uma versão modificada do PIM-SM, chamada PIM-SSM. O protocolo PIM-SSM simplifica o IP Multicast, mas não permite sua implantação gradual.

4.1.1 O Protocolo EXPRESS

O protocolo EXPRESS (EXPlicitely REquested Single Source multicast) [36] foi projetado para otimizar a distribuição multicast para uma única fonte. O protocolo EXPRESS propõe a extensão do modelo IP Multicast com a abstração de "canal". Um canal multicast é um serviço de distribuição multicast, de fonte única, identificado pelo par (S, E), onde S é o endereço da fonte de dados e E é um endereço de destino do canal. Apenas a fonte S é autorizada a enviar dados no canal (S, E). Ao se conectar a um canal, um membro recebe apenas os dados transmitidos por S para E. Dois canais, (S, E) e (S', E) não possuem qualquer relação, apesar do endereço de destino em comum. A árvore de distribuição EXPRESS é construída pelo ECMP (*Express Count Management Protocol*) [36], um protocolo de gerenciamento que além de manter a árvore, suporta a contagem de membros por fonte e votação.

O protocolo foi especialmente projetado para aplicações de tipo assinatura, usando canais lógicos, como TV na Internet ou distribuição de arquivos. Embora tenha sido elaborado para aplicações de fonte única, aplicações com múltiplas fontes podem ser construídas utilizando-se vários canais, ou permitindo que várias fontes utilizem o mesmo canal utilizando um mecanismo de mais alto nível para que todas as fontes passem pela mesma estação-fonte do canal.

A proposta EXPRESS é importante por representar uma solução simples para o problema de alocação de endereços multicast. A concatenação de um endereço unicast com um endereço IP Classe D (utilizado como E na identificação do canal) proporciona um identificador único para o canal multicast, uma vez que o endereço unicast é único, por definição. Além disso, a abstração de canal simplifica, intrinsecamente, alguns problemas de gerenciamento de grupo, como o controle de acesso ao envio de dados. O protocolo ECMP (*Express Count Management Protocol*) acrescenta mecanismos para a contagem de membros do grupo. O modelo de canal proposto no EXPRESS foi adotado pelo IETF, dando origem a um novo serviço dentro do IP Multicast, denominado *Source-Specific Multicast* (SSM). O protocolo PIM-SSM implementa o serviço SSM, porém não incorpora qualquer mecanismo de gerenciamento de grupo adicional.

4.1.2 O Serviço *Source-Specific Multicast*

A proposta do EXPRESS motivou a implementação do serviço SSM (*Source-Specific Multicast*) na Internet. O modelo de canal introduzido pelo EXPRESS simplifica bastante a arquitetura multicast, porque a alocação de endereços multicast passa a ser um problema local à fonte de dados, e porque a distribuição multicast pode ser implementada por um protocolo de roteamento mais simples.

No IPv4, o serviço SSM é implementado pelo protocolo PIM-SSM (*Protocol Independent Multicast - Source Specific Multicast*) [37] e pelo protocolo IGMPv3 [17]. A versão 3 do protocolo IGMP é necessária, pelo mecanismo de filtragem de fontes presente apenas nesta versão. Na verdade, o mecanismo de filtragem possui mais funcionalidade do que seria necessário para implementar o serviço SSM. O protocolo PIM-SSM é uma versão modificada (simplificada) do protocolo PIM-SM, com novas regras de tratamento de mensagens e com a capacidade de interpretar corretamente as mensagens IGMPv3 enviadas pelas estações.

O PIM-SSM considera a existência de uma única fonte por canal multicast, e constrói *exclusivamente* árvores de distribuição por fonte, em vez de árvores compartilhadas como o PIM-SM. Desta forma, não há necessidade de roteadores de *rendez-vous* (RPs) para gerenciar a árvore, nem de mecanismos de mapeamento de endereços multicast para RPs. Mecanismos de descoberta de fontes no inter-domínio, como o protocolo MSDP, também não são necessários. Por outro lado, o PIM-SSM supõe que os receptores potenciais conhecem o endereço da fonte, sendo assim capazes de gerar pedidos IGMPv3 transportando este endereço. O mecanismo de descoberta de fontes utilizado na prática está fora do escopo da definição do PIM-SSM. A descoberta pode ser realizada através de serviços como o correio eletrônico, publicação na web, ou pelo protocolo SAP

(*Session Announcement Protocol*) [38].

O IANA alocou uma faixa de endereços multicast exclusiva para a utilização do serviço SSM, a faixa `232/8` (`232.0.0.0` à `232.255.255.255`). Assim, é possível a coexistência do serviço específico à fonte (implementado pelo protocolo PIM-SSM na faixa exclusiva) com o serviço IP Multicast tradicional (utilizando o PIM-SM, fora da faixa exclusiva). Para tanto, as implementações do PIM-SM devem ser modificadas, tanto nos roteadores de borda (BR – *Border Router*) quanto nos roteadores de *backbone*. Além disso, o protocolo IGMPv3 deve ser implementado em todas as estações e roteadores de primeiro-salto. Ainda assim, o modelo SSM obteve uma grande aceitação da comunidade de pesquisa, uma vez que é mais simples e elimina algumas das limitações da arquitetura de distribuição multicast PIM-SM/MBGP/MSDP.

O serviço SSM baseado no PIM-SSM e IGMPv3

Quando um receptor deseja se conectar a um canal multicast, ele envia um pedido IGMPv3 ao seu roteador DR, especificando o par de endereços (S, G). Se o endereço G pertence à faixa exclusiva do serviço SSM, o roteador DR envia uma mensagem $join(S, G)$ na direção da fonte, de forma a ser adicionado à árvore de distribuição do canal (S, G). Se ao invés disso o endereço G está fora da faixa SSM, o roteador terá o comportamento normal do protocolo PIM-SM, ou seja, enviando uma mensagem $join(*, G)$ em direção ao RP. Após a recepção do primeiro pacote da fonte, através da árvore compartilhada, o DR pode eventualmente se conectar à árvore por fonte, enviando uma mensagem $join(S, G)$. Neste caso, porém, o receptor não se beneficiará das vantagens do serviço SSM. O DR receberá os dados enviados por qualquer fonte ao grupo G através da árvore compartilhada. Se o DR não implementar a filtragem de fontes, todos os dados serão encaminhados ao receptor. A Figura 17 mostra os componentes do modelo SSM no IPv4.

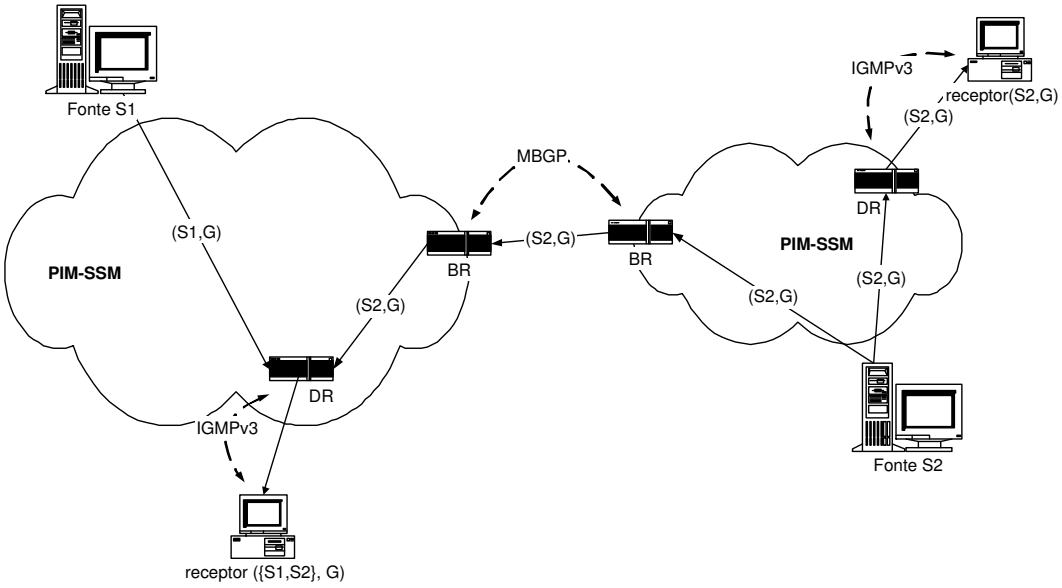


Figura 17: O modelo de distribuição PIM-SSM/IGMPv3.

Para garantir a coexistência e a compatibilidade entre os protocolos PIM-SM e PIM-SSM, as fontes PIM-SM devem utilizar apenas endereços multicast fora da faixa exclusiva SSM, enquanto as fontes SSM devem se limitar a esta faixa. O comportamento dos roteadores designados e dos roteadores de *rendez-vous* deve ser modificado, portanto. Os roteadores DR devem enviar uma mensagem $join(S, G)$ diretamente, sem passar pelo RP, se o endereço multicast G é um endereço SSM. Ao mesmo tempo, os RPs não devem aceitar um pedido $join(*, G)$ na faixa exclusiva SSM

e não devem se apresentar como candidatos (no processo de eleição de RPs) para endereços na faixa SSM. Fontes não devem enviar dados para o RP, se o endereço destino corresponde a um canal multicast. Ao invés disso, elas devem utilizar a árvore por fonte diretamente.

4.1.3 O Protocolo REUNITE

O protocolo REUNITE (REcursive UNIcast TrEes) [39] introduziu a técnica de unicast recursivo para a distribuição multicast. A idéia é utilizar a infra-estrutura de roteamento unicast para realizar a distribuição multicast. De maneira semelhante aos protocolos EXPRESS e PIM-SSM, o REUNITE constrói apenas árvores por fonte. Um grupo é identificado pelo par (endereço unicast da fonte, número de porta) e pacotes de dados possuem endereços destino unicast.

A motivação inicial do REUNITE foi a observação de que em árvores multicast típicas, a maioria dos roteadores simplesmente reenvia os pacotes recebidos de uma interface de entrada através de uma *única* interface de saída. Em outras palavras, os roteadores de ramificação da árvore são minoria (Figura 18). Estas características já haviam sido observadas por Pansiot e Grad [40] e foram confirmadas mais recentemente por Chalmers e Almeroth [41]. Apesar disso, em todos os protocolos multicast IP, todos os roteadores atravessados pela árvore de distribuição armazenam informação por grupo (ou por fonte/grupo).

A idéia no REUNITE foi então dividir o estado de roteamento em duas tabelas, uma tabela de controle (*Multicast Control Table* – MCT) e uma tabela de reenvio de pacotes (*Multicast Forwarding Table* – MFT). A MCT é armazenada no plano de controle dos roteadores, enquanto que a MFT é instalada no plano de dados. Apenas nós de ramificação da árvore possuem entradas na tabela de reenvio, MFT, outros roteadores apenas guardam informação na tabela de controle, MCT, que não é consultada durante o envio de *dados*. Os roteadores de ramificação utilizam a informação armazenada na MFT para realizar a duplicação de pacotes corretamente.

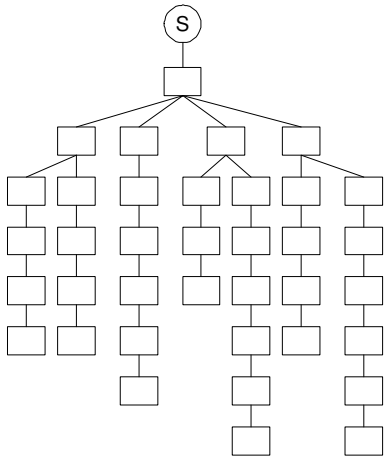


Figura 18: Forma de uma árvore multicast típica.

O REUNITE identifica uma conversação multicast por um par $<S, P>$, onde S é o endereço unicast da fonte e P é um número de porta, escolhido pela fonte. O REUNITE não utiliza endereços multicast IP Classe D. À medida que os receptores se conectam à árvore, o protocolo preenche as tabelas MCT e MFT de forma a construir a árvore. Os estados armazenados nas tabelas são voláteis (*soft-state*). O REUNITE utiliza dois tipos de mensagens para construção e manutenção da árvore: *join* e *tree*. Mensagens *join* são periodicamente enviadas pelos receptores na direção da fonte (S). Já as mensagens *tree* são periodicamente enviadas em multicast pela fonte, para atualizar o estado volátil da árvore. Apenas os nós de ramificação para o grupo $<S_1, P_1>$ mantém entradas $<S_1, P_1>$ em sua tabela MFT. A tabela de controle, MCT, não é

usada no encaminhamento de dados. Roteadores de simples reenvio (que não realizam bifurcação na árvore) possuem entradas MCT para $<S_1, P_1>$, mas nenhuma entrada MFT para este grupo.

A distribuição multicast através de unicast recursivo

A idéia de base da técnica de unicast recursivo é utilizar endereços de destino *unicast* nos pacotes de dados. Roteadores que atuam como nós de ramificação da árvore de um determinado grupo são responsáveis pela criação de cópias dos pacotes de dados. Estas cópias possuem seus endereços de destino modificados, de forma a que todos os membros do grupo recebam uma cópia da informação.

A Figura 19 ilustra a distribuição de dados no protocolo REUNITE. A fonte envia os dados endereçados ao primeiro membro que se conectou ao grupo. Em um nó de ramificação, R_n, os pacotes recebidos possuem como endereço de destino o endereço do primeiro receptor, r_i, que se conectou ao grupo na sub-árvore abaixo de R_n. O receptor r_i é armazenado em uma entrada especial da MFT, MFT$<S>.dst$. O roteador R_n cria uma cópia dos dados para cada receptor presente em sua MFT (o endereço de destino de cada cópia é igual ao endereço unicast do receptor). A cópia original do pacote é reenviada a r_i. Neste exemplo, S produz pacotes de dados endereçados a r_1 (estes pacotes chegam a r_1 inalterados). O roteador R_1 cria uma cópia do pacote com endereço de destino r_4. O roteador R_3 simplesmente reenvia os pacotes sem precisar consultar sua MFT. O roteador R_5 cria uma cópia do pacote para r_8 e finalmente R_7 cria cópias para r_5 e r_6.

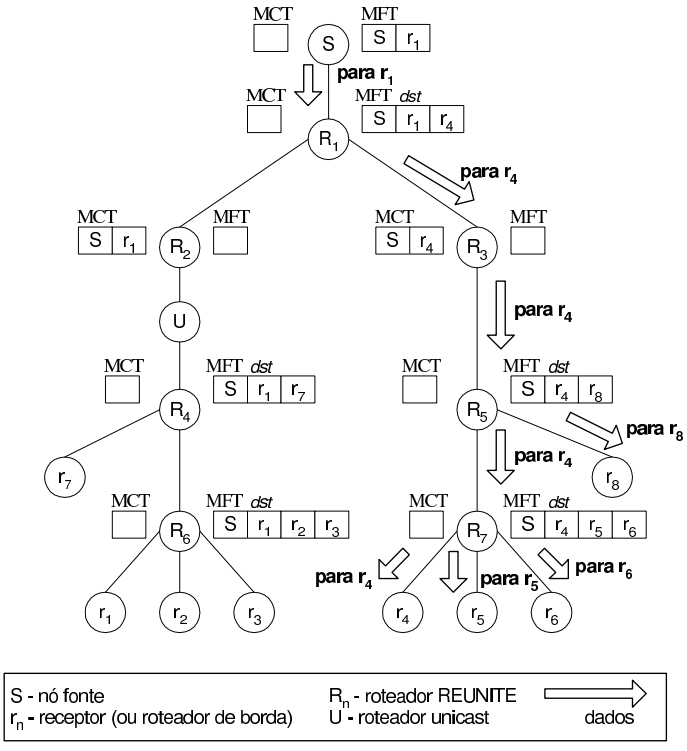


Figura 19: Árvore de distribuição REUNITE.

A técnica de unicast recursivo permite a implementação progressiva do serviço multicast porque o reenvio de dados é baseado em endereços unicast. Desta forma, roteadores que não implementam o multicast são suportados de forma transparente. Estes roteadores são incapazes de funcionar como nós de ramificação da árvore, mas podem, no entanto, reenviar os pacotes sem problemas, uma vez que estes são endereçados em unicast.

A Figura 20 ilustra o mecanismo de construção da árvore REUNITE com um exemplo onde este falha na construção da SPT. Considere as rotas unicast: r_1>R_2>R_1>S ; S>R_1>R_3>r_1 ; r_2>R_3>R_1>S ; S>R_4>r_2. Suponha os seguintes eventos: r_1 se conecta a $<S,P>$, r_2 se conecta a $<S,P>$ e r_1 deixa o grupo.

O receptor r_1 "assina" o canal através do envio de um $join(S,r_1)$[1] para S. Esta mensagem atinge S uma vez que não existe estado para este canal nos roteadores. Diz-se que r_1 se conectou ao canal $<S,P>$ em S. A fonte S começa então a produzir mensagens $tree(S,r_1)$ que são enviadas a r_1 (em unicast). As mensagens *tree* instalam *soft-state* para $<S,P>$ nos roteadores pelos quais elas passam. Os roteadores R_1 e R_3 criam uma entrada $<S,r_1>$ em suas MCTs. Em seguida, r_2 se conecta ao canal. O $join(S,r_2)$ trafega na direção de S atingindo a árvore em R_3. O roteador R_3 descarta a mensagem $join(S,r_2)$, cria uma MFT$<S>$ com *dst* igual a r_1, adiciona r_2 à MFT$<S>$ e remove $<S,r_1>$ de sua MCT. O roteador R_3 se torna um nó de ramificação e vai conseqüentemente produzir mensagens $tree(S,r_2)$ quando da recepção de $tree(S,r_1)$. Diz-se que r_2 se conectou ao canal em R_3. Pacotes de dados enviados para o canal (endereçados para r_1) são duplicados em R_3 e endereçados a r_2. Mensagens *join* subseqüentes enviadas por r_1 e r_2 atualizam o *soft-state* das entradas respectivas nas MFTs de S e R_3.

Nesta configuração, r_1 recebe os dados de S através do caminho mais curto, mas não r_2. Uma vez que as rotas unicast entre S e r_2 são assimétricas e como R_3 intercepta o $join(S,r_2)$, os dados seguem o caminho S>R_1>R_3>r_2, o mesmo caminho utilizado pelas mensagens *tree* para irem da fonte S a r_2 (Figura 20(a)).

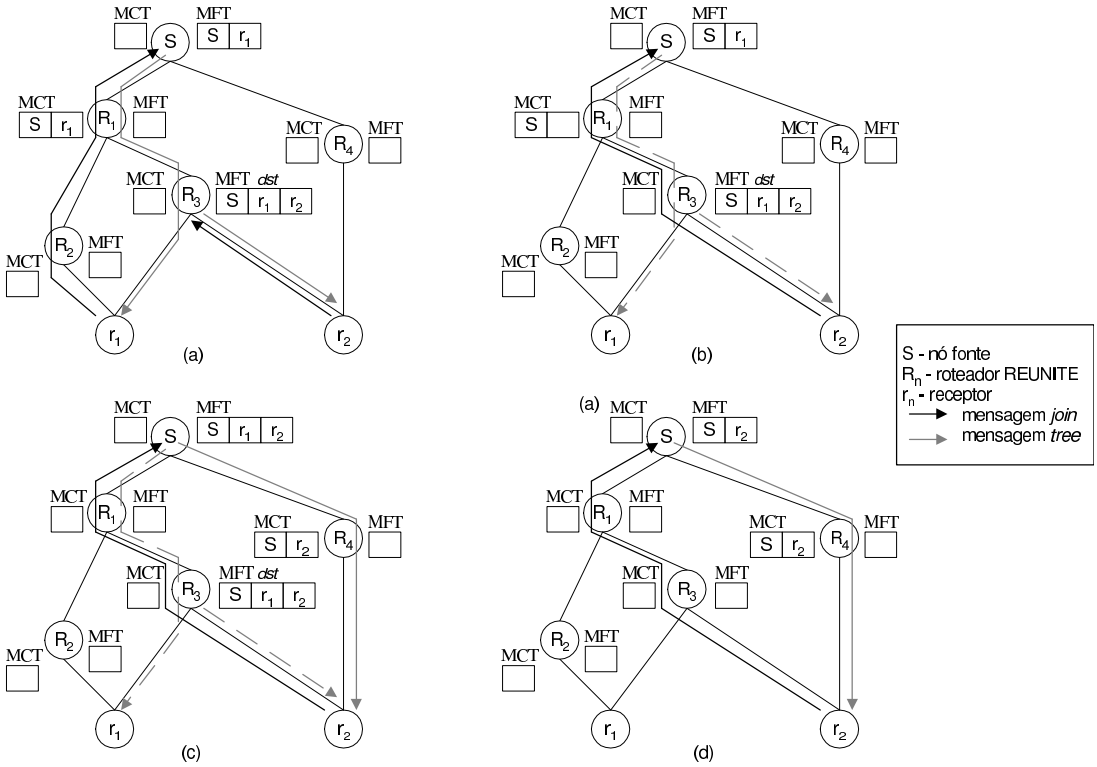


Figura 20: O mecanismo de construção da árvore REUNITE.

Os estados mantidos na MCT e MFT são *soft-state*. Os receptores periodicamente enviam mensagens $join(S,r_i)$ e a fonte periodicamente produz uma mensagem $tree(S,r_i)$ em multicast. Para se desconectar do canal o receptor deve simplesmente parar o envio de mensagens *join*. Quando a árvore está estabilizada, os $tree(S,r_i)$ atualizam o *soft-state* de entradas r_i nas MCTs

[1]No resto desta seção, o termo $<S>$ pode ser usado no lugar de $<S,P>$ referindo-se ao canal multicast.

dos roteadores assim como as entradas MFT$<S>.dst = r_i$. Os $join(S, r_j)$ atualizam a entrada r_j na MFT do nó onde r_j se conectou a $<S>$. Na Figura 20, os $join(S, r_1)$ atualizam r_1 na MFT de S e os $join(S, r_2)$ atualizam r_2 na MFT de R_3.

Suponha agora que r_1 deixa o grupo, parando de emitir $join(S, r_1)$. Como a entrada r_1 na MFT de S deixa de ser atualizada, após a expiração do temporizador *t1* a entrada r_1 se torna *stale* (desatualizada). Um segundo temporizador, *t2*, é criado e vai destruir a entrada r_1 caso esta não seja mais atualizada. Uma vez que r_1 está *stale*, S envia mensagens $tree(S, r_1)$ *marcadas* (Figura 20(b)). Mensagens $tree(S, r_1)$ marcadas significam que o fluxo de dados endereçados a r_1 pode cessar em breve, logo a parte da árvore contendo r_1 nas tabelas de roteamento deve ser reconfigurada. As MFT nos nós de ramificação que possuem MFT$<S>.dst = r_1$ tornam-se *stale* após a recepção das mensagens *tree* marcadas. Em nós de simples reenvio, a recepção de $tree(S, r_1)$ marcadas causa a destruição de entradas r_1 da MCT. Conseqüentemente, os $join(S, r_2)$ deixam de ser interceptados por R_3 (porque sua MFT está *stale*) e atingem S. Desta forma, r_2 agora se conecta ao canal $<S, P>$ em S (Figure 20(c)). Algum tempo depois, *t2* irá expirar, acarretando a retirada de r_1 das MFTs de S e R_3. Como R_3 pára de receber mensagens *tree*, sua MFT$<S>$ é destruída (Figura 20(d)). Agora, r_2 recebe os dados através do caminho mais curto a partir de S.

O roteamento assimétrico pode levar o REUNITE a produzir cópias desnecessárias de pacotes em certos enlaces.[2] A Figura 21 mostra um exemplo. O primeiro receptor, r_1, envia um $join(S, r_1)$ que segue o caminho r_1>R_4>R_2>R_1>S. As mensagens $tree(S, r_1)$ seguem a rota S>R_1>R_6>R_4>r_1. Suponha agora que r_2 se conecta e que o $join(S, r_2)$ passa por r_2>R_5>R_3>R_1>S. Os $tree(S, r_1)$ (produzidos por S) e os $tree(S, r_2)$ (criados em R_1) atravessam ambos o enlace R_1-R_6. Como R_6 não recebe mensagens *join* destes receptores, ele não se identifica como nó de ramificação. S produz pacotes de dados endereçados a r_1, em seguida R_1 cria cópias endereçadas a r_2. Desta forma duas cópias de cada pacote atravessam o enlace R_1-R_6.

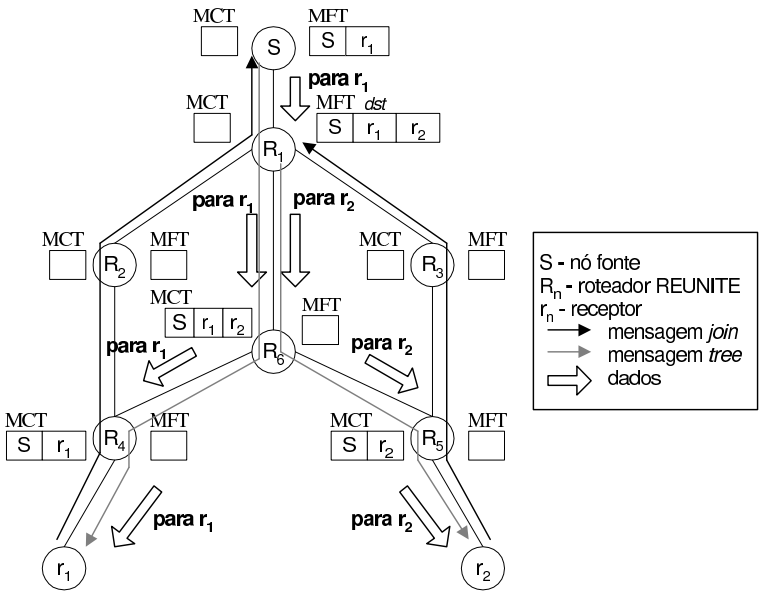


Figura 21: Duplicação de pacotes devido a rotas assimétricas no REUNITE.

Como conseqüência, o custo (número de cópias do mesmo pacote nos enlaces da rede) de uma árvore REUNITE pode ser maior que o custo de uma árvore por fonte (*source tree*) construída

[2]Esta possibilidade também existe para redes contendo roteadores puramente unicast ou quando um roteador REUNITE está sobrecarregado. Em ambos os casos, o nó de ramificação migrará para um roteador não ideal podendo acarretar a duplicação de pacotes. Consultar [39] para uma descrição detalhada.

por um protocolo tradicional como PIM-SM (*Protocol Independent Multicast - Sparse Mode*)[30], uma vez que a técnica RPF (*Reverse Path Forwarding*) garante que no máximo uma cópia de cada pacote trafegará por cada enlace da rede.

4.1.4 O Protocolo HBH

O protocolo HBH (*Hop-By-Hop multicast routing protocol*) [42] utiliza a técnica de unicast recursivo, como o REUNITE, mas adota a abstração de canal do EXPRESS e SSM. Como o REUNITE, o HBH suporta de forma transparente a presença de roteadores unicast puros. No entanto, o HBH constrói árvores SPT (*shortest-path tree*) ao invés de árvores SPT reversas, como a maioria dos outros protocolos. O algoritmo de construção da árvore evita a duplicação desnecessária de pacotes em cenários onde o roteamento unicast é assimétrico.

O HBH possui um algoritmo de construção de árvores capaz de tratar os casos patológicos devidos às assimetrias do roteamento unicast. O HBH utiliza duas tabelas, MCT e MFT, que possuem aproximadamente a mesma função que no REUNITE. A diferença é que cada entrada nas tabelas de HBH armazena o endereço do *próximo nó de ramificação* em vez do endereço dos *receptores finais* (exceto no roteador de ramificação mais próximo do receptor). A MFT não possui entrada *dst*. A Figura 22 ilustra a distribuição multicast em uma árvore HBH. Os pacotes de dados recebidos por um roteador de ramificação, H_n, possuem endereço de destino unicast igual a H_n (no REUNITE os dados são endereçados a MFT.<*dst*>). Esta diferença de concepção torna a estrutura da árvore HBH mais estável que a REUNITE, já que no HBH as entradas correspondentes a receptores estão localizadas próximas às folhas da árvore. O HBH identifica o canal multicast através do par $<S, G>$, onde S é o endereço unicast da fonte e G um endereço IP classe D alocado pela fonte. Isto evita o problema de alocação de endereços multicast e mantém a compatibilidade com IP Multicast. Desta forma, o HBH pode suportar nuvens IP Multicast como folhas da árvore de distribuição.

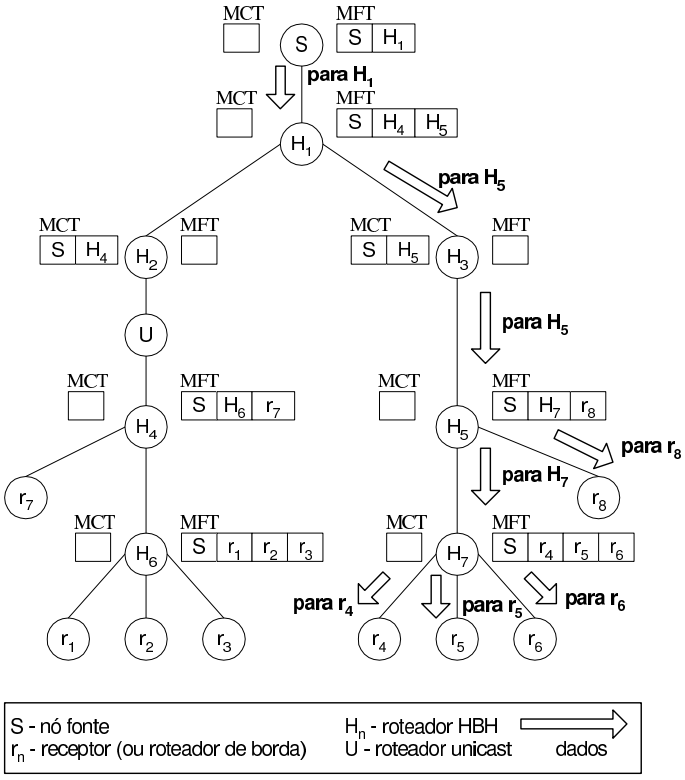


Figura 22: Árvore de distribuição HBH.

A estrutura da árvore HBH possui a vantagem de maior estabilidade das entradas nas tabelas de roteamento que o REUNITE. A contrapartida é que no HBH cada pacote recebido por um nó de ramificação produz $n+1$ cópias modificadas enquanto em REUNITE n cópias são produzidas. O mecanismo de gestão da árvore de HBH reduz o impacto da saída de um membro na estrutura da árvore. Isto é possível porque a entrada correspondente a um receptor é localizada o mais próximo possível deste receptor no HBH.

Gestão da árvore HBH

O protocolo HBH utiliza três tipos de mensagem: *join*, *tree* e *fusion*. Mensagens *join* são periodicamente enviadas pelos receptores na direção da fonte e atualizam o estado de reenvio (entradas na MFT) no roteador onde o receptor se conectou ao canal. Um nó de ramificação "se conecta" ele próprio ao canal, no próximo nó de ramificação na direção da fonte. Desta forma, as mensagens *join* podem ser interceptadas por nós de ramificação que em seguida enviam mensagens *join* assinadas por eles próprios. A fonte periodicamente envia uma mensagem *tree* em multicast sobre a árvore que é responsável por atualizar o resto da estrutura da árvore. Mensagens *fusion* são enviadas por nós de ramificação em potencial e participam na construção da árvore em conjunto com as mensagens *tree*.

Cada nó HBH na árvore de distribuição de S possui uma MCT<S> *ou* uma MFT<S>. Um nó de simples reenvio possui uma MCT<S> com uma única entrada à qual dois temporizadores são associados, *t1* e *t2*. Quando *t1* expira, a MCT torna-se *stale*, sendo destruída após a expiração de *t2*.

Um nó de ramificação da árvore de S possui uma MFT<S>. Dois temporizadores, *t1* e *t2*, são associados a cada entrada na MFT<S>. Quando *t1* expira a entrada torna-se *stale*, sendo destruída após a expiração de *t2*. No HBH, uma entrada *stale* é usada pra o reenvio de dados, mas não causa a produção de mensagens *tree*. Uma entrada na MFT<S> no HBH pode também estar *marcada*. Uma entrada marcada é usada no reenvio de mensagens *tree*, mas não no reenvio de dados. Em [42] é apresentada uma descrição detalhada das regras de processamento das mensagens HBH. As idéias básicas são: o primeiro *join* enviado por um receptor nunca é interceptado, chegando à fonte; mensagens *tree* são periodicamente enviadas em multicast pela fonte; estas são combinadas com mensagens *fusion* enviadas por nós de ramificação em potencial de forma a construir e refinar a estrutura da árvore.

Considere novamente o exemplo da Seção 4.1.3 para mostrar a construção da árvore HBH. A Figura 23 retoma o cenário da Figura 20. O receptor r_1 se conecta ao canal em S, que começa o envio de mensagens $tree(S, r_1)$. Estas mensagens criam uma MCT<S> contendo r_1 nos nós H_1 e H_3 (Figura 20(a)). Quando r_2 se conecta ao canal enviando o primeiro $join(S, r_2)$, este não é interceptado chegando a S (o primeiro *join* nunca é interceptado). O $tree(S, r_2)$ produzido pela fonte cria uma MCT<S> em H_4 (Figura 23(b)). Ambos os receptores estão conectados à fonte através do caminho mais curto (fonte receptor).

Suponha agora que r_3 (rotas unicast: S>H_1>H_3>r_3 e r_3>H_3>H_1>S) se conecta ao canal. r_3 envia um $join(S, r_3)$ para S, que começa a enviar mensagens $tree(S, r_3)$. Como H_1 recebe duas mensagens *tree* diferentes, este envia um $fusion(S, r_1, r_3)$ na direção da fonte. A recepção do *fusion* faz com que S *marque* as entradas r_1 e r_3 e inclua H_1 na MFT<S>. Da mesma forma que H_1, H_3 recebe os $tree(S, r_1)$ e $tree(S, r_3)$ e envia então uma mensagem $fusion(S, r_1, r_3)$ para a fonte (Figura 23(c)). A MFT de H_3 agora contém r_1 e r_3. Os $join(S, r_1)$ subseqüentes são interceptados por H_1 e atualizam a entrada r_1 (marcada) na MFT de H_1. Os $join(S, r_3)$ atualizam a entrada r_3 na MFT de H_3. A distribuição de dados se passa da seguinte forma. A fonte S envia pacotes endereçados a H_1, que os reenvia endereçados a H_3. O roteador H_3 cria cópias que são enviadas a r_1 e r_3. Subseqüentemente, como S deixa de receber os $join(S, r_1)$ e $join(S, r_3)$, as entradas correspondentes em sua MFT são destruídas. A estrutura estabilizada

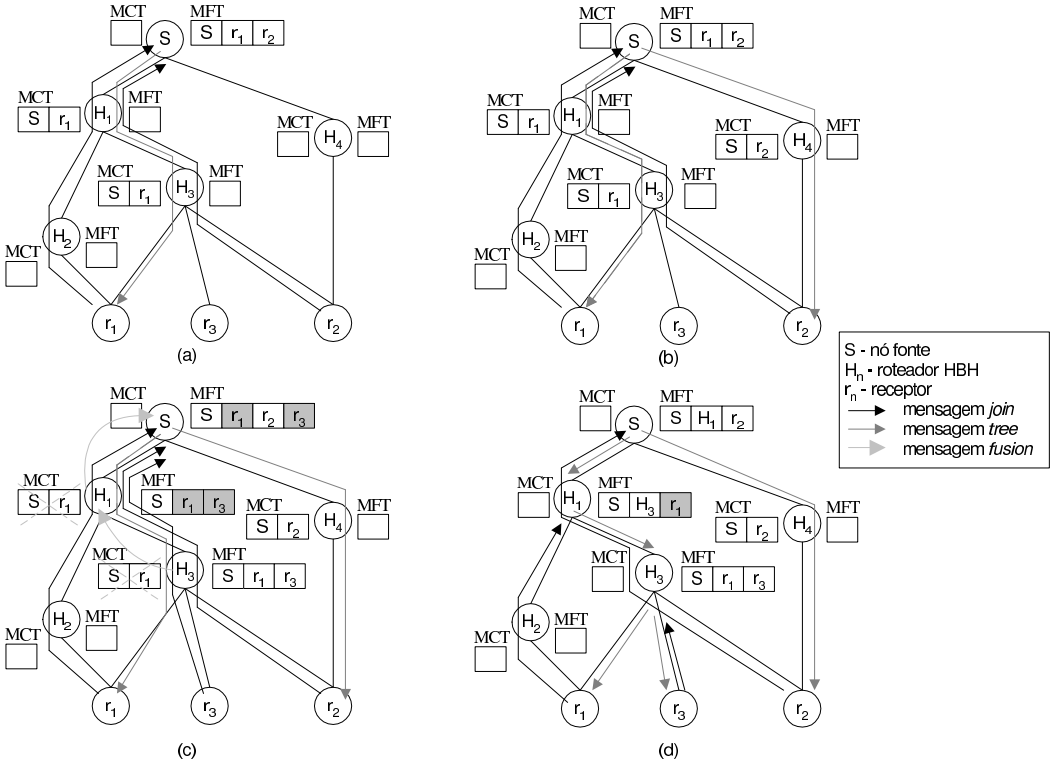


Figura 23: O mecanismo de construção da árvore HBH.

da árvore é mostrada na Figura 23(d). Desta forma, o HBH utiliza o nó de ramificação ideal para a distribuição multicast. O problema apresentado na Figura 21 é resolvido através do envio de um $fusion(S, r_1, r_2)$ de H_6 para a fonte, de maneira equivalente ao exemplo desta seção.

4.2 Protocolos de Roteamento Especializados

Outros protocolos de roteamento que se baseiam em um serviço de comunicação de grupo específico foram propostos. Em geral, estes protocolos são especializados para um determinado tipo de aplicação ou tecnologia de rede.

Os protocolos XCAST [43] e DCM (*Distributed Core Multicast*) [44] foram projetados para evitar o problema de escalabilidade em termos da quantidade de grupos multicast ativos. Estes protocolos tentam diminuir a quantidade de estados por grupo armazenada pelos roteadores. O XCAST transporta uma lista explícita de receptores dentro de um novo cabeçalho IP (no caso do IPv4) ou de uma extensão de roteamento (no caso do IPv6). Cada roteador deve examinar este cabeçalho para descobrir se este roteador é um ponto de ramificação da árvore. Em caso afirmativo, o roteador cria uma cópia do pacote para cada ramo da árvore, com as correspondentes listas dos receptores alcançáveis a partir de cada interface de saída. O XCAST é escalável em termos de grupos ativos, pois não armazena estado por grupo nos roteadores. Por outro lado, o tamanho do grupo é limitado uma vez que a lista explícita dos receptores é transportada nos pacotes de dados.

O DCM utiliza uma estratégia diferente. Vários roteadores *core* (*Distributed Core Routers* – DCRs) são distribuídos na periferia da rede. Os DCRs utilizam um protocolo específico para o gerenciamento dos membros dos grupos multicast. Cada rede local possui um DCR responsável pelo envio de tráfego aos receptores locais. O ganho de escalabilidade do DCM está no fato de apenas os roteadores DCR armazenarem informação sobre os grupos multicast ativos.

Alguns protocolos foram propostos para o caso específico das redes ad hoc. O AMRoute [45, 46] é um sistema implementado especializado para redes ad hoc. O AMRoute primeiro constrói uma topologia virtual em malha sobre a qual é construída uma árvore de distribuição compartilhada. O ODMRP (*On-Demand Multicast Routing Protocol*) [47] é outro protocolo multicast para redes ad hoc. No ODMRP, a fonte de dados é responsável por construir uma malha interligando os receptores através da qual são inundados os dados. O MAODV [48] constrói uma árvore compartilhada sob demanda, utilizando o mesmo tipo de mecanismo de pedido/resposta implementado pelo AODV para o tráfego unicast.

4.3 Redes Multicast Virtuais

Devido à lenta implantação do serviço multicast, diferentes propostas surgiram para um serviço de comunicação de grupo alternativo [49]. Em geral, estas propostas consideram que a camada de rede, ou não implementa o serviço de distribuição multicast, ou o implementa apenas em algumas regiões. Desta maneira, a maioria das soluções propostas contrói *overlays* (redes virtuais) que implementam a funcionalidade de multicast *sobre* a camada IP. Estes *overlays* podem ser implementados por roteadores ou pelas estações finais. No primeiro caso, a maior parte das propostas pertencem à categoria da geração automática ded túneis. O segundo tipo de solução pode ser classificado como multicast aplicativo. Nesta categoria, a própria aplicação implementa o serviço de comunicação de grupo de que neceessita, ou algumas vezes um *middleware* é implementado e provê o serviço a diferentes aplicações.

Recentemente, a geração automática de túneis multicast vem sendo discutida no IETF. A geração automática de túneis consite em um programa, ou protocolo, que configura de maneira automatizada túneis multicast.

Um dos mecanismos de tunelamento utilizados atualmente para interconectar duas redes multicast isoladas é o UMTP (UDP *Multicast Tunneling Protocol*) [50]. O UMTP encapsula os datagramas multicast dentro de datagramas unicast UDP, o que possibilita sua implementação como um processo de nível usuário nas estações finais. Os túneis UMTP conectam uma estação mestre, que periodicamente envia mensagens *join-group*, a uma estação escravo.

Diferentes mecanismos foram propostos no IETF para automatizar a geração de túneis UMTP. Finlayson et al. [51] propuseram que roteadores multicast também pudessem ser uma das extremidades nos túneis UMTP. Neste caso, e assumindo que o serviço SSM seja utilizado, as mensagens UMTP *join-group* são endereçadas à fonte SSM (em vez de a uma extremidade de túnel explícita) e interceptadas pelo primeiro roteador multicast encontrado na direção da fonte. Um número de porta UDP especial deve ser reservado, para permitir que o roteador multicast identifique a mensagem UMTP *join-group*. Liefooghe e Goossens [52] propuseram um mecanismo de localização dinâmica de servidores de túneis. A escolha de um servidor adequado leva em conta, primeiramente, a proximidade do cliente, de acordo com uma hierarquia onde os servidores são classificados de acordo com sua posição geográfica. Além disso, um mecanismo de caracterização de caminhos é proposto para que se possa escolher um servidor entre os diferentes servidores de um mesmo nível geográfico. O cliente constrói túneis UMTP provisórios com todos os servidores possíveis, em seqüência. O cliente envia tráfego em um canal de teste sobre este túnel, este tráfego é reenviado sobre o túnel pelo servidor. Um mecanismo de controle de fluxo por taxa é implementado sobre este canal de teste, e as medidas obtidas compõem uma métrica utilizada para escolher o melhor servidor de túneis.

O AMT (*Automatic Multicast Tunnelling*) [53] é um protocolo alternativo que não se baseia no UDP, podendo transportar outros protocolos. O AMT propõe um mecanismo semelhante ao "6to4" [54], proposto para a construção de túneis IPv6 sobre redes IPv4. Desta forma, o AMT trata a "Internet unicast" (a nuvem de roteadores que não falam multicast), como um enlace não-broadcast de múltiplo acesso (NBMA – *Non-Broadcast Multi-Access link*). O AMT implementa pseudo-interfaces de rede que são utilizadas como rota *default* para o tráfego multicast. Um

Site AMT (por exemplo, uma rede que implementa o multicast, mas que não está conectada à infra-estrutura multicast nativa de uma rede *backbone*) possui um *Gateway* AMT (uma estação ou roteador) que implementa uma pseudo-interface AMT. O *Gateway* AMT se comunica com um roteador *Relay* AMT, que está conectado à infra-estrutura multicast nativa. O protocolo se baseia em um prefixo *anycast* especial para anunciar a rota para um roteador *Relay* AMT disponível na infra-estrutura unicast. O tráfego multicast é encapsulado em UDP entre os roteadores *Relay* AMT e o *Gateway* AMT. Por *default*, todo o tráfego multicast produzido na infra-estrutura multicast nativa será encaminhado para o *Site* AMT. Para evitar o reenvio de todo o tráfego, os autores do AMT propõe a utilização do IGMP como protocolo de sinalização. O AMT permite a recepção de tráfego multicast no *Site* AMT, e possibilita que uma fonte no *Site* AMT emita tráfego.

A arquitetura LAR (*Logical Addressing and Routing architecture*) proposta por Pansiot *et al.* [55] se diferencia dos mecanismos de geração automática de túneis. O LAR propõe a utilização de uma estrutura de endereçamento lógica em cima da estrutura de endereçamento IP padrão. A idéia é que as estações possuam endereços lógicos, como os nomes de domínio utilizados no serviço DNS (*Domain Name Service*). Os endereços lógicos são utilizados para identificar estações móveis, estações com múltiplas conexões à rede (*multi-homed*), e grupos multicast. No caso do multicast, o DNS contém uma associação entre o nome do grupo, o endereço do grupo, e o endereço do gerente do grupo. Quando uma estação deseja se conectar a um grupo multicast, ela deve aprender a partir do DNS qual é a estação gerente responsável por este grupo. O gerente de grupo controla o acesso de emissores e receptores ao grupo. A construção da árvore de distribuição se baseia em mensagens *join* e *join acknowledgement*, que transitam em direções opostas, desta forma o LAR é capaz de implementar árvores SPT, como o protocolo HBH. No entanto, a estrutura LAR se propõe a substituir o modelo de grupo do IP Multicast, enquanto que o HBH adota o modelo SSM.

4.4 Multicast Aplicativo

Neste tipo de sistema, as estações finais são responsáveis por criar uma topologia de distribuição no nível aplicação, algumas vezes misturando o roteamento multicast e unicast. Existem diferentes motivações para implementar o multicast aplicativo. Em geral, esta solução proporciona maior controle sobre a árvore criada; serviços adicionais podem ser facilmente implantados; e estações que não suportam o multicast nativo são suportadas. Porém, não se pode atingir a mesma eficiência na utilização dos recursos da rede proporcionada pelo uso do multicast nativo.

O multicast aplicativo é também conhecido por "multicast baseado na estação" (*Host-based Multicast*). As estações finais (ou, algumas vezes, nós bem conhecidos, como por exemplo um roteador de saída) e auto-configuram para criar uma topologia de distribuição com múltiplos nós, às vezes misturando os roteamentos multicast e unicast. Por exemplo, onde o multicast é a técnica mais eficiente (como no caso de múltiplas estações localizadas em um segmento Ethernet), uma área multicast é mantida. Em outras situações, quando o roteamento unicast é mais adequado, ou a única opção (por exemplo no roteamento inter-domínio), túneis unicast são criados. Existem diferentes motivações para o multicast aplicativo:

- oferecer controle total da árvore na topologia virtual;
- funcionalidades adicionais podem ser facilmente implementadas (por ex., utilizar comunicação confiável sobre um enlace com fortes perdas, no caso de um caminho com roteadores freqüentemente congestionados, etc.);
- a inclusão de nós móveis que não implementam o roteamento multicast padrão;
- a inclusão de áreas bem específicas (por ex., uma rede ad hoc onde os nós se comunicam sem nenhuma infra-estrutura) que se baseiam em protocolos de roteamento multicast dedicados;

- a inclusão de nós que não possuem acesso a nenhum serviço multicast.

Diversos sistemas de multicast aplicativo foram propostos [49]. A maneira como a distribuição é realizada sobre a topologia virtual é um dos fatores que diferenciam os sistemas.

A construção da árvore pode ser realizada de forma centralizada ou distribuída. Entre os sistemas centralizados, há os que supõem conhecimento total (por ex. HBM – *Host-Based Multicast* [56]) ou parcial da topologia (por ex. ALMI – *Application Level Multicast Infrastructure* [57]).

Entre os sistemas distribuídos, há os que constroem diretamente uma árvore de distribuição, e outros que primeiro constroem uma topologia em malha. Por exemplo, o Narada utiliza como topologia virtual uma malha completa entre as estações finais [58]. Sobre esta malha, o Narada constrói uma árvore usando o algoritmo *Reverse Path Forwarding* (RPF). Outros sistemas que primeiro constroem uma malha podem ser encontrados em [59, 60].

Já o Yoid, proposto em [61] é um mecanismo de multicast aplicativo que constrtói a árvore diretamente. O Yoid utiliza um ponto de *rendez-vous* que fornece informações sobre a sessão multicast e realiza algumas funções de gerenciamento e sinalização. Em [62] é descrito o protocolo de gerenciamento de árvores do Yoid, o YMTP (*Yoid Tree management Protocol*). A arquitetura do Yoid é bastante complexa pois considera outros mecanismos além do roteamento, como por exemplo o transporte confiável. Outros exemplos de protocolos da categoria de implementação direta da árvore são o TBCP (*Tree Building Control Protocol*) [63] e HMTP (*Host Multicast Tree Protocol*) [64].

5 O Serviço Multicast no IPv6

O IP Multicast deve ser uma das tecnologias chave na implantação da nova geração da Internet, com o protocolo IPv6. Uma vez que no IPv4 o multicast foi introduzido como uma extensão, nem todos os nós IPv4 implementam o serviço multicast. Por outro lado, a especificação do IPv6 requer que *todos* os nós suportem o multicast. Como consequência imediata, uma implementação IPv6 não tem que suportar túneis multicast, usados no IPv4 para interconectar as ilhas multicast.

Embora as noções básicas do IP Multicast, como o modelo de serviço, sejam comuns ao IPv4 e ao IPv6, algumas características novas são introduzidas no multicast IPv6. Enquanto no IPv4 o escopo de um grupo multicast é na maioria das vezes especificado utilizando o campo TTL do pacote de dados, o IPv6 limita de maneira explícita o escopo dos pacotes através de um campo fixo do endereço multicast. A arquitetura de endereçamento IPv6 [65] incorpora o escopo dentro dos endereços unicast e multicast. Um campo de 4 bits é reservado para a definição do escopo nos endereços multicast. Os roteadores não encaminham pacotes multicast cujo escopo específico está fora do domínio dentro do qual o endereço multicast é válido. Outra diferença importante com relação ao IPv4, é que muitas das implementações do multicast IPv4 utilizam endereços unicast para identificar uma interface de rede, isto não é adequado no IPv6. Um nó IPv6 pode associar múltiplos endereços a uma única interface, o que poderia provocar erros de configuração.

Portanto, um endereço multicast IPv6 identifica um grupo de interfaces (geralmente em nós distintos). Os endereços multicast IPv6 possuem o formato mostrado na Figura 24.

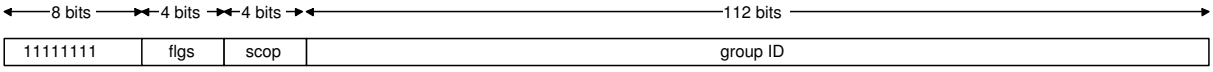


Figura 24: Formato do endereço multicast IPv6.

Os primeiros 8 bits com valor 1 no início do endereço IPv6 identificam este endereço como sendo um endereço multicast. A arquitetura de endereçamento IPv6 na sua especificação atual [66]

define apenas o último dos 4 bits de *flags*, chamado T. Os 3 primeiros bits dos *flags* são reservados e devem utilizar o valor 0. O bit $T = 0$ indica um endereço multicast alocado permanentemente ("well-known address") [67], atribuído pelo IANA (*Internet Assigned Numbers Authority*). O bit $T = 1$ indica um endereço alocado temporariamente (ou endereço transiente). O campo `scop`, de 4 bits, transporta um valor de escopo multicast, usado para limitar o alcance de um grupo multicast.

O IETF está atualmente trabalhando em uma extensão da arquitetura de endereçamento IPv6 que permite a alocação de endereços multicast baseada nos prefixos de endereços unicast [68]. A idéia é poder delegar endereços multicast ao mesmo tempo que os prefixos unicast, desta forma os operadores de rede podem identificar suas faixas de endereços multicast sem a necessidade de um protocolo de alocação de endereços inter-domínio. O formato de endereço atualmente proposto é mostrado na Figura 25.

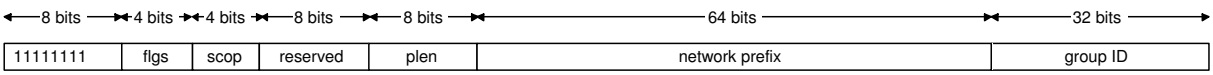


Figura 25: Formato do endereço multicast IPv6 baseado no prefixo unicast.

Neste novo formato, um segundo *flag*, P, foi introduzido no campo `flgs`. O bit $P = 1$ indica um endereço multicast alocado com base no prefixo de rede, senão, $P = 0$. Além disso, se $P = 1$, então T deve possuir valor igual a 1. O campo `plen` especifica o comprimento do campo de prefixo de rede que identifica a sub-rede. O campo `network prefix` identifica o prefixo de rede da sub-rede unicast à qual o endereço é atribuído, e o último campo, `group ID`, é um identificador de grupo de 32 bits.

Alguns dos protocolos descritos anteriormente, como o PIM-SM, já possuem implementações para o IPv6 com as adaptações necessárias. Para o suporte do serviço SSM, existem ainda algumas questões em discussão, como a reserva de uma faixa de endereços exclusiva. Uma faixa de endereços SSM pode ser obtida a partir de endereços multicast alocados com base no prefixo unicast, sendo atualmente discutida no IETF [68]. Outros protocolos, como o MSDP e o MBGP, ainda não foram tratados para o IPv6. O MSDP provavelmente não existirá no IPv6. Suas chances de ser implementado diminuem com o sucesso do serviço SSM, e com o desenvolvimento do protocolo BGMP.

Um componente essencial do serviço SSM no IPv4 é o protocolo IGMPv3. No IPv6, a funcionalidade do IGMP é incluída no protocolo ICMP (*Internet Control Message Protocol*) [69]. A primeira versão IPv6 do IGMP foi chamada MLD (*Multicast Listener Discovery*) [70]. A **versão 1** do MLD implementa a mesma funcionalidade que a **versão 2** do IGMP, e corresponde à tradução do IGMPv2 para a semântica do IPv6. O *Multicast Listener Discovery* é utilizado por roteadores IPv6 para descobrir a presença de "ouvintes" multicast nos enlaces diretamente conectados ao roteador (ou seja, estações interessadas em receber pacotes multicast). A versão seguinte, MLDv2, introduz a filtragem de fontes implementada pelo IGMPv3 no IPv4 [71]. Desta forma, o roteador pode conhecer os grupos de interesse de suas estações clientes, assim como o conjunto de fontes que estes clientes desejam escutar. O protocolo está em fase final de padronização pelo IETF.

Referências

[1] C. Diot, W. Dabbous e J. Crowcroft, "Multipoint communication: A survey of protocols, functions and mechanisms", *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 277–290, abril 1997.

[2] M. Handley e J. Crowcroft, "Internet multicast today", *The Internet Protocol Journal*, vol. 2, no. 4, pp. 2–19, dezembro 1999.

[3] I. P. 802, *IEEE P802.1p: Supplement to MAC Bridges: Traffic Class Expediting and Dynamic Multicast Filtering.* IEEE, Incorp. in IEEE Standard 802.1D, Part 3: Media Access Control (MAC) Bridges: Revision, 1998.

[4] V. Fuller, T. Li, J. Yu e K. Varadhan, *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy.* RFC 1519, setembro 1993.

[5] B. Williamson, *Developing IP Multicast Networks.* Cisco Press, janeiro 2000.

[6] M. Handley, D. Thaler e R. Kermode, *Multicast-Scope Zone Announcement Protocol (MZAP).* RFC 2776, fevereiro 2000.

[7] D. Meyer e P. Lothberg, *GLOP Addressing in 233/8.* RFC 2770, fevereiro 2000.

[8] D. Thaler, M. Handley e D. Estrin, *The Internet Multicast Address Allocation Architecture.* RFC 2908, setembro 2000.

[9] S. Hanna, B. Patel e M. Shah, *Multicast Address Dynamic Client Allocation Protocol (MADCAP).* RFC 2730, dezembro 1999.

[10] R. Droms, *Dynamic Host Configuration Protocol.* RFC 1531, outubro 1993.

[11] M. Handley e S. Hanna, *Multicast Address Allocation Protocol (AAP)*, junho 2000. Work in progress, <draft-ietf-malloc-aap-04.txt>.

[12] P. Radoslavov, D. Estrin, R. Govindan, M. Handley, S. Kumar e D. Thaler, *The Multicast Address-Set Claim (MASC) Protocol.* RFC 2909, setembro 2000.

[13] Y. Rekhter e T. Li, *A Border Gateway Protocol 4 (BGP-4).* RFC 1771, março 1995.

[14] S. Kumar, P. Radoslavov, D. Thaler, C. Alaettinoglu, D. Estrin e M. Handley, "The MASC/BGMP architecture for inter-domain multicast routing", in *ACM SIGCOMM'98*, pp. 93–104, setembro 1998.

[15] S. Deering, *Host Extensions for IP Multicasting.* RFC 1112, agosto 1989.

[16] W. Fenner, *Internet Group Management Protocol, Version 2.* RFC 2236, novembro 1997.

[17] B. Cain, S. Deering, I. Kouvelas, B. Fenner e A. Thyagarajan, *Internet Group Management Protocol, Version 3.* RFC 3376, outubro 2002.

[18] L. Sahasrabuddhe e B. Mukherjee, "Multicast routing algorithms and protocols: a tutorial", *IEEE Network*, pp. 90–102, janeiro 2000.

[19] B. Wang e J. C. Hou, "Multicast routing and its QoS extension: Problems, algorithms, and protocols", *IEEE Network*, pp. 22–36, janeiro 2000.

[20] J. Moy, *OSPF version 2.* RFC 2328, abril 1998.

[21] K. C. Almeroth, "The evolution of multicast: From the MBone to interdomain multicast to Internet2 deployment", *IEEE Network*, pp. 10–20, janeiro 2000.

[22] D. Waitzman, C. Partridge e S. Deering, *Distance Vector Multicast Routing Protocol.* RFC 1075, novembro 1988.

[23] G. Malkin, *RIP Version 2.* RFC 2453, novembro 1998.

[24] C. Huitema, *Routing in the Internet.* Prentice Hall, 2nd ed., 1999.

[25] J. Moy, *Multicast Extensions to OSPF.* RFC 1584, março 1994.

[26] J. Moy, *MOSPF: Analysis and Experience.* RFC 1585, março 1994.

[27] T. Ballardie, P. Francis e J. Crowcroft, "Core based trees (CBT)", in *SIGCOMM'93 - Communications Architectures, Protocols and Applications*, pp. 85–95, 1993.

[28] A. Ballardie, *Core Based Trees (CBT) Multicast Routing Architecture.* RFC 2201, setembro 1997.

[29] S. Deering, D. L. Estrin, D. Farinacci, V. Jacobson, C.-G. Liu e L. Mei, "The PIM architecture for wide-area multicast routing", *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, pp. 153–162, abril 1996.

[30] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma e L. Wei, *Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification.* RFC 2362, junho 1998.

[31] I. Brown, J. Crowcroft, M. Handley e B. Cain, "Internet multicast tomorrow", *The Internet Protocol Journal*, vol. 5, no. 4, pp. 2–19, dezembro 2002.

[32] T. Bates, Y. Rekhter, R. Chandra e D. Katz, *Multiprotocol Extensions for BGP-4.* RFC 2858, junho 2000.

[33] D. M. (Editor) e B. F. (Editor), *Multicast Source Discovery Protocol (MSDP).* Work in progress, <draft-ietf-msdp-spec-14.txt>, novembro 2002.

[34] D. Thaler, *Border Gateway Multicast Protocol (BGMP): Protocol Specification.* Work in progress, <draft-ietf-bgmp-spec-03.txt>, junho 2002.

[35] C. Diot, B. N. Levine, B. Liles, H. Kassem e D. Balensiefen, "Deployment issues for the IP multicast service and architecture", *IEEE Network*, pp. 78–88, janeiro 2000.

[36] H. W. Holbrook e D. R. Cheriton, "IP multicast channels: EXPRESS support for large-scale single-source applications", in *ACM SIGCOMM'99*, setembro 1999.

[37] S. Bhattacharyya, C. Diot, L. Giuliano, R. Rockell, J. Meylor, D. Meyer, G. Shepherd e B. Haberman, *An Overview of Source-Specific Multicast (SSM).* Work in progress, <draft-ietf-ssm-overview-04.txt>, novembro 2002.

[38] M. Handley, C. Perkins e E. Whelan, *Session Announcement Protocol.* RFC 2974, outubro 2000.

[39] I. Stoica, T. S. E. Ng e H. Zhang, "REUNITE: A recursive unicast approach to multicast", in *IEEE INFOCOM'2000*, março 2000.

[40] J.-J. Pansiot e D. Grad, "On routes and multicast trees on the Internet", *ACM Computer Communication Review*, vol. 28, no. 1, pp. 41–50, janeiro 1998.

[41] R. C. Chalmers e K. C. Almeroth, "Modeling the branching characteristics and efficiency gains in global multicast trees", in *IEEE INFOCOM'2001*, abril 2001.

[42] L. H. M. K. Costa, S. Fdida e O. C. M. B. Duarte, "Hop by hop multicast routing protocol", in *ACM SIGCOMM'2001*, pp. 249–259, agosto 2001.

[43] R. Boivie, N. Feldman, Y. Imai, W. Livens, D. Ooms e O. Paridaens, *Explicit Multicast (Xcast) Basic Specification.* Work in progress, <draft-ooms-xcast-basic-spec-04.txt>, janeiro 2003.

[44] L. Blazevic e J.-Y. L. Boudec, "Distributed core multicast (dcm): a multicast routing protocol for many groups with few receivers", *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 5, pp. 6–21, outubro 1999.

[45] M. Liu, R. Talpade e A. McAuley, "Amroute: Adhoc multicast routing protocol", Technical Report TR 99-1, CSHCN, 1999.

[46] Bommaiah, A. McAuley, R. Talpade e M. Liu, *AMRoute: Adhoc multicast routing protocol*, agosto 1998. work in progress; <draft-manet-amroute-00.txt>.

[47] S.-J. Lee, W. Su e M. Gerla, "On-demand multicast routing protocol in multihop wireless mobile networks", *ACM/Baltzer Mobile Networks and Applications*, vol. 7, no. 6, pp. 441–453, dezembro 2002.

[48] E. M. Royer e C. E. Perkins, "Multicast operation of the ad-hoc on-demand distance vector routing protocol", in *Mobile Computing and Networking*, pp. 207–218, agosto 1999.

[49] A. El-Sayed, V. Roca e L. Mathy, "A survey of proposals for an alternative group communication service", *IEEE Network*, vol. 17, no. 1, pp. 46–51, janeiro 2003.

[50] R. Finlayson, *The UDP Multicast Tunneling Protocol.* Work in progress, <draft-finlayson-umtp-07.txt>, setembro 2002.

[51] R. Finlayson, R. Perlman e D. Rajwan, *Accelerating the Deployment of Multicast Using Automatic Tunneling.* Work in progress, <draft-finlayson-mboned-autotunneling-00.txt>, fevereiro 2001.

[52] P. Liefooghe e M. Goossens, "An architecture for seamless access to multicast content", in *IEEE Conference on Local Computer Networks*, novembro 2000.

[53] D. Thaler, M. Talwar, L. Vicisano e D. Ooms, *IPv4 Automatic Multicast Without Explicit Tunnels.* Work in progress, <draft-ietf-mboned-auto-multicast-01.txt>, abril 2002.

[54] B. Carpenter e K. Moore, *Connection of IPv6 Domains via IPv4 Clouds.* RFC 3056, fevereiro 2001.

[55] J.-J. Pansiot, A. Alloui, T. Noel e D. Grad, "A new architecture for sparse-mode inter-domain multicasting", in *EUNICE Open European Summer School*, setembro 2000.

[56] A. El-Sayed e V. Roca, "A host-based multicast (HBM) solution for group communications", in *1st IEEE International Conference on Networking*, julho 2001.

[57] D. Pendarakis, S. Shi, D. Verma e M. Waldvogel, "ALMI: An application level multicast infrastructure", in *3rd USENIX Symposium on Internet Technologies and Systems*, pp. 49–60, março 2001.

[58] Y. Chu, S. Rao e H. Zhang, "A case for end system multicast", in *ACM SIGMETRICS*, junho 2000.

[59] S. Zhuang, B. Zhao, A. Joseph, R. Katz e J. Kubiatowicz, "Bayeux: An architecture for scalable and fault-tolerant wide-area data dissemination", in *International Workshop on Network and Operating System Support for Digital Audio and Video*, junho 2001.

[60] J. Liebeherr, M. Nahas e W. Si, "Application-layer multicast with delaunay triangulatios", in *IEEE GLOBECOM*, novembro 2001.

[61] P. Francis, "Yoid: extending the multicast internet architecture". Unrefered Report; http://www.aciri.org/yoid/, setembro 1999.

[62] P. Francis, "Yoid tree management protocol (ytmp) specification". Unrefered Report; http://www.aciri.org/yoid/, dezembro 1999.

[63] L. Mathy, R. Canonico e D. Hutchinson, "An overlay tree building control protocol", in *International Workshop on Networked Group Communication*, novembro 2001.

[64] B. Zhang, S. Jamin e L. Zhang, "Host multicast: A framework for delivering multicast to end users", in *IEEE INFOCOM*, junho 2002.

[65] R. Hinden e S. Deering, *IP Version 6 Addressing Architecture.* RFC 2373, julho 1998.

[66] R. Hinden e S. Deering, *IP Version 6 Addressing Architecture*, outubro 2002. Work in progress, <draft-ietf-ipngwg-addr-arch-v3-13.txt>.

[67] R. Hinden e S. Deering, *IPv6 Multicast Address Assignments.* RFC 2375, julho 1998.

[68] B. Haberman e D. Thaler, *Unicast-Prefix-based IPv6 Multicast Addresses.* RFC 3306, agosto 2002.

[69] A. Conta e S. Deering, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification.* RFC 2463, dezembro 1998.

[70] S. Deering, W. Fenner e B. Haberman, *Multicast Listener Discovery (MLD) for IPv6.* RFC 2710, novembro 1999.

[71] R. Vida, L. Costa, S. Fdida, S. Deering, B. Fenner, I. Kouvelas e B. Haberman, *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*, outubro 2002. Work in progress, <draft-vida-mld-v2-05.txt>.