Universidade Federal da Paraíba

Centro de Ciências e Tecnologia

Curso de Pós-Graduação em Informática

Dissertação de Mestrado

Avaliação de Protocolos *Multicast* em Redes TCP-IP

Artur Henrique Kronbauer

Campina Grande, Paraíba, Brasil

Universidade Federal da Paraíba

Centro de Ciências e Tecnologia

Curso de Pós-Graduação em Informática

Artur Henrique Kronbauer

Avaliação de Protocolos *Multicast* em Redes TCP-IP

Dissertação submetida ao Curso de Pós-Graduação em Informática do Centro de Ciências e Tecnologia da Universidade Federal da Paraíba, em cumprimento às exigências para obtenção do grau de Mestre em Informática.

Área de Concentração: Ciência da Computação
Sub-Área: Redes de Computadores

Orientadores: Joberto Sérgio B. Martins, Dr.
Maria Izabel Cavalcanti Cabral, D.Sc

Campina Grande, Junho de 1998

K93a Kronbauer, Artur Henrique.
Avaliação de protocolos Multicast em redes TCP-IP / Artur Henrique Kronbauer. - Campina Grande, 1998.
109 f.

Dissertação (Mestrado em Informática) - Universidade Federal da Paraíba, Centro de Ciências e Tecnologia, 1998.
"Orientação : Prof. Dr. Joberto Sérgio Barbosa Martins, Profa. Dra. Maria Izabel Cavalcanti Cabral".
Referências.

1. Redes de Computadores. 2. Protocolos Multicast. 3. Redes TCP-IP. 4. Dissertação - Informática. I. Martins, Joberto Sérgio Barbosa. II. Cabral, Maria Izabel Cavalcanti. III. Universidade Federal da Paraíba - Campina Grande (PB). IV. Título

CDU 004.7(043)

AVALIAÇÃO DE PROTOCOLOS MULTICAST EM REDES TCP-IP

ARTUR HENRIQUE KRONBAUER

DISSERTAÇÃO APROVADA EM 15.06.1998

PROF. JOBERTO SÉRGIO BARBOSA MARTINS, Dr.
Orientador

PROFa. MARIA IZABEL CAVALCANTI CABRAL, D.Sc
Orientadora

PROF. JOSÉ NEUMAN DE SOUZA, Dr.
Examinador

PROF. PEDRO SÉRGIO NICOLLETTI, M.Sc
Examinador

CAMPINA GRANDE - PB

a meus pais Arthur e Ruth Kronbauer, pela ajuda, confiança e amor que sempre depositaram em mim.

AGRADECIMENTOS

• Agradeço a meus pais e irmãos que mesmo nas horas mais difíceis me deram força, amor e compreensão.

• Agradeço ao Professor Joberto e a Professora Maria Izabel pelo apoio e confiança.

• Agradeço aos meus amigos Marcos Wagner, Patrícia Andrade, Milena Pessoa, Paulo Sausen e Paulo Martins, que tantas vezes participaram de momentos alegres e tristes ao meu lado.

• Agradeço a Valter Rosa e Anna Dolores e demais colegas de trabalho da Telebahia, que contribuem para o bom andamento de minhas atividades profissionais.

• Finalmente, agradeço aos muitos amigos que fiz por todos os lugares que passei e que, de uma forma ou de outra, contribuíram para a realização deste trabalho: gaúchos, baianos, paraibanos, goianos, pernambucanos, maranhenses e alagoanos.

RESUMO

Com o desenvolvimento de novas aplicações em redes de computadores, baseadas na família de protocolos TCP/IP, surge a necessidade de melhorias para dar uma resposta efetiva às novas exigências impostas. Os estudos e pesquisas voltados às redes TCP/IP procuram a adequação de novas tecnologias de comunicação, que viabilizem a transferência segura e veloz da informação e o oferecimento de novos serviços que exigem, por exemplo, o transporte de diferentes mídias. Para isso, investiga-se novos mecanismos que possam reduzir ao máximo a ocupação da banda passante. Nesse contexto, estão sendo desenvolvidas novas formas de transferência de dados para as redes de computadores que conduzem ao denominado tráfego *multicast*.

Este trabalho apresenta um estudo dos protocolos *multicast* no que diz respeito às formas de roteamento. Nele é proposto o SPM (Simulador de Protocolo *Multicast*) desenvolvido para permitir a análise de performance dos protocolos *multicast* MOSPF (*Multicast Extensions Open Shortest Path First*), CBT (*Core Based Trees*) e PIM-SM (*Protocol Independent -Multicast - Sparce Mode*). Os resultados obtidos na simulação desses protocolos possibilitam a análise e a escolha daqueles que mais se adaptem às necessidades de uma rede TCP/IP específica, de acordo com sua topologia e recursos disponíveis.

ABSTRACT

With the development of new computer network applications based on the TCP/IP family protocols, it arises the necessity of improvements in order to give an effective response to the new requirements. The studies and researches focused on TCP/IP networks try to adequate the new communication technologies, that make viable the fast and secure information transmission, and the offer of new services that demand, for example, the transportation of different media. For this, it has been investigated new mechanisms that can reduce the use of the bandwidth as much as possible. In this context, it has been developed new forms of data transference for the computer network that lead to the denominated *multicast* traffic.

This dissertation presents a study of multicast protocols regarding the routing forms. It is proposed the MPS (Multicast Protocol Simulator) developed to allow the performance analysis of the multicast protocols MOSPF (*Multicast Extensions Open Shortest Path First*), CBT (*Core Based Trees*) and PIM-SM (*Protocol Independent -Multicast - Sparce Mode*). The results gotten in the simulation of these protocols make possible the analysis and choice of the ones that best fit to the necessity of a specific TCP/IP network, according to its topology and the resources available to it.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

CAPÍTULO 1

INTRODUÇÃO

Com o avanço das tecnologias de comunicação e processamento de informações, os usuários ficam cada vez mais exigentes quanto aos recursos informatizados que são providos, exigindo assim que melhores serviços sejam oferecidos pelas redes de computadores. Dessa forma, a família de protocolos TCP/IP [COMER 91], que é largamente utilizada em *Intranets*, *Extranets* e na *Internet*, está sofrendo evoluções para assegurar serviços mais elaborados que exigem, por exemplo, recursos multimídia.

Uma das principais diferenças a nível de requisição de rede entre os serviços tradicionais e os que estão sendo desenvolvidos atualmente, diz respeito à restrição imposta ao atraso dos pacotes. Por exemplo, o tráfego multimídia propõe a transferência de som, imagem e dados interativamente, limitando o tempo para a geração dos pacotes e a sua absorção por parte dos destinatários.

Outro ponto importante a salientar em relação ao tráfego multimídia, em particular, é que se espera que o mesmo pacote seja requisitado por mais de um receptor. Desta forma, soluções estão sendo propostas para que se evite a sobrecarga dos enlaces, diminuindo assim

congestionamentos nas redes e atrasos fim-a-fim [NORO 94A].

Atualmente, temos três técnicas de transferência de informações entre computadores numa rede IP: o *unicast*, o *broadcast* e o *multicast* [COMER 91].

A comunicação *unicast* especifica o fluxo de informação em direção a um único destino, sendo preestabelecido o endereço do receptor no cabeçalho do pacote IP. Esta técnica é largamente utilizada hoje, mas não apresenta um bom desempenho para algumas das novas aplicações, pois não se mostra flexível para possibilitar que vários destinos recebam os mesmos pacotes sem que estes tenham que ser gerados para cada um deles.

Por outro lado, temos a comunicação *broadcast*, onde uma origem envia um fluxo de informações que será recebido por todos os *hosts* na rede. Esta técnica é de extrema utilidade nos casos em que uma informação deva ser distribuída a todos os *hosts* e roteadores em uma determinada rede, porém deve ser evitada nos demais casos, já que inunda todas as interfaces da rede gerando tráfego desnecessário.

A comunicação *multicast* se enquadra entre as duas técnicas descritas acima, pois o fluxo de informações gerado por uma determinada origem, atinge a um ou mais *hosts* associados a um grupo de destino e resolve os problemas apresentados pelo *unicast* e *broadcast*, uma vez que ele gera somente o tráfego necessário para atingir os destinos desejados, fazendo com que não transitem pacotes idênticos pelo mesmo enlace e replicando-os somente quando necessário.

Alguns dos ganhos e finalidades da comunicação *multicast* são: facilitar o trabalho do *host* que está enviando um pacote para vários destinatários, aliviar a sobrecarga dos *hosts* que não precisam receber pacotes, dar oportunidade a um *host* de se integrar a um ou mais grupos *multicast* e aliviar a sobrecarga dos enlaces que estão transferindo as informações [RFC1112].

A implementação da técnica *multicast* exige funções básicas que devem ser incorporadas às estações (computadores e roteadores) em uma rede TCP/IP. Elas são incorporadas através de protocolos desenvolvidos para fazer o gerenciamento dos grupos *multicast* e o roteamento das informações na rede.

Os trabalhos pioneiros com relação ao IP *multicast* foram lançados no final dos anos 80, por Steve Deering [RFC 1075] [RFC 1112], sendo atualmente recomendados pela IETF (*Internet Engineering Task Force*), órgão que define a padronização relacionada com as redes TCP/IP. As primeiras aplicações foram desenvolvidas no MBONE (*Multicast Backbone*) para transmissões de áudio e vídeo geradas nas reuniões da IETF.

Nos dias de hoje, estão sendo desenvolvidos ou aperfeiçoados protocolos como o DVMRP (*Distance Vector Multicast Routing Protocol*) [PUSAT 96], o MOSPF [RFC 1584], o CBT [RFC 2189], o PIM-SM [RFC 2117] e o PIM-DM (*Protocol Independent Multicast - Dense Mode*) [HELMY 97], para determinar o roteamento das comunicações *multicast*; enquanto que a gerência dos grupos *multicast* é feita pelo IGMP (*Internet Group Management Protocol*) [FENNE 97], com a finalidade de suportar a comunicação aos grupos de *hosts*, usando exclusivamente a classe D de endereçamento IP, para identificar um específico grupo *multicast*.

1.1. OBJETIVOS

Os objetivos desse trabalho são os seguintes:

1. Realizar pesquisas e estudos sobre a transferência de informações via a técnica *multicast* incluindo os principais conceitos, definições, protocolos e arquitetura.

2. Estudar as características dos protocolos de roteamento *multicast*, abrangendo a filosofia de implementação de cada um.

3. Propor e implementar um simulador que possa servir como ferramenta de análise para os protocolos de roteamento *multicast*. Nesse contexto, foi construído o Simulador de Protocolos *Multicast* (SPM) voltado para a simulação de modelos de redes TCP/IP que podem suportar os protocolos de roteamento *multicast*.

4. Apresentar uma avaliação inicial dos protocolos de roteamento MOSPF, PIM-SM e CBT, inseridos no ambiente do *backbone* da RNP (Rede Nacional de Pesquisa).

1.2. RELEVÂNCIA

Nas últimas décadas várias pesquisas foram desenvolvidas com a intenção de otimizar a utilização da banda passante nas redes de computadores. Estes estudos criaram a concepção da transferência de informações via a técnica *multicast*, dando início a construção de vários protocolos de roteamento que pudessem absorver esta filosofia. Estes protocolos embora alguns ainda em desenvolvimento, começam a ser implementados nas redes de computadores, fazendo-se necessário a análise dos mesmos.

Vários trabalhos têm sido desenvolvidos para identificar e avaliar as possíveis melhorias a serem implementadas nos protocolos de roteamento *multicast* [NORO94A] [NORO94B]. O SPM, proposto nessa dissertação, é uma ferramenta que pode dar suporte à comunidade TCP/IP nos estudos de análise de desempenho de protocolos *multicast*.

O seu projeto, por ser modular e extensível, permite que estudos de avaliação de desempenho de novos protocolos possam ser feitos, a partir da versão do SPM aqui apresentada.

1.3. ESTRUTURA DA DISSERTAÇÃO

O restante dessa dissertação está organizado em 6 (seis) capítulos:

- No capítulo 2 é apresentado o estado da comunicação *multicast*;

- No capítulo 3 são abordados todos os protocolos de roteamento envolvidos nas transações *multicast*;

- No capítulo 4 é apresentado o SPM e mostrado como ele pode simular redes TCP/IP com roteamento *multicast*;

- No capítulo 5 é apresentado o cenário para as simulações, abrangendo um estudo de caso baseado na RNP;

- No capítulo 6 são apresentados os resultados das simulações realizadas, e

- No capítulo 7 são apresentadas as conclusões e sugestões de continuidade da presente dissertação.

CAPÍTULO 2

COMUNICAÇÃO *MULTICAST*

Vamos supor que somos analistas de sistemas de uma grande empresa, e temos que atualizar o *software* das *workstations* de aproximadamente 2.000 engenheiros durante o final de semana. Assim, na manhã de sábado, damos a partida em um sistema de distribuição de *software* e começamos a execução da atividade.

Notamos, entretanto, que o servidor que fazia a atualização estava estabelecendo muitas conexões uma a uma com as *workstations*, e simplesmente, não havia largura de banda suficiente na rede para lidar com tantas conexões ao mesmo tempo, de modo que a transferência pára [HURW 97].

Caso estivéssemos usando transações *multicast* isso ocorreria? E como podemos implementar este tipo de transação em nossas redes?

Essas perguntas, que representam uma vertente do estado da arte em comunicação de alto desempenho, são respondidas neste capítulo após: uma rápida definição do modelo IP *multicast* (seção 2.1); a descrição dos algoritmos de roteamento (seção 2.2); a definição de QoS (Qualidade de Serviço) e os protocolos utilizados para prover este recurso (seção 2.3) e, finalmente, a apresentação do MBONE (seção 2.4).

2.1. O MODELO IP *MULTICAST*

O IP *multicast* é definido como a transmissão de um pacote IP para um grupo de *hosts*, utilizando o mesmo método *best-effort* usado em transmissões *unicast*, com a diferença de que o pacote originalmente gerado sofrerá replicações quando necessário, para possibilitar a recepção por todos os integrantes do grupo ao qual o pacote foi endereçado.

A associação de um host a um grupo é dinâmica, isto é, eles podem juntar-se ou deixar o grupo conforme seu interesse, independente da localização ou do número de *hosts* associados, além disso, podem ser membros de um ou mais grupos de acordo com suas necessidades.

Um grupo é identificado por um único endereço de destino, podendo ser permanente ou transitório. Um grupo permanente pode ter membros ou não em um determinado momento sem que seja extinto. Já o transitório, só existe enquanto os associados estiverem conectados.

O protocolo de transporte usado para comunicações *multicast* é o UDP (*User Datagram Protocol*) [RFC 0768], pois este não implementa estabelecimento de conexão, o que é propício à arquitetura *multicast*, que não pode ser orientada à conexão devido às transferências serem de um *host* para um grupo e não para um outro *host*, o que se caracterizaria como ponto-a-ponto.

Este modelo se estende para a definição de endereçamento *multicast*, gerência de grupos e roteamento dos pacotes, que será tratado nos próximos tópicos.

2.1.1. ENDEREÇOS *MULTICAST*

Os grupos *multicast* são identificados pela classe D de endereçamento IP, a qual possui os quatro primeiros bits de ordem mais alta definidos com "1110", seguidos por 28 bits que identificam os grupos, desta forma variam do endereço 224.0.0.0 a 239.255.255.255.

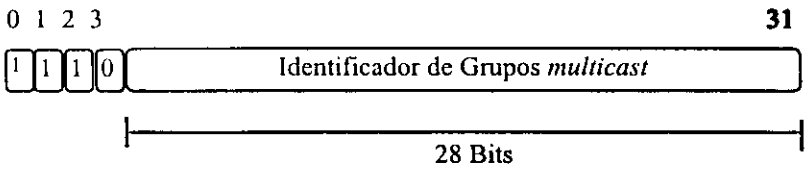


Figura 1. Formato da classe D de endereçamento IP

O mapeamento dos endereços *multicast* IP em endereços da camada MAC IEEE-802 é realizado a partir do endereço 01-00-5E (hex), o qual é o endereço reservado para multicast na tecnologia IEEE-802, sendo passado os 23 bits de baixa ordem do endereço IP, que representam o grupo *multicast*, para os 23 bits de baixa ordem do endereço da tecnologia sendo utilizada (*Ethernet*, *Token Ring*, etc).

De acordo com a Figura 2, podemos observar que cinco bits do endereço IP são ignorados após o mapeamento, possibilitando assim que 32 endereços *multicast* IP diferentes possam ser mapeados dentro do mesmo endereço *Ethernet*. Por exemplo, o endereço *multicast* 224.138.8.5 (E0-8A-08-05) e o endereço 225.10.8.5 (E1-0A-08-05) serão mapeados no mesmo endereço *Ethernet* (01-00-5E-0A-08-05).

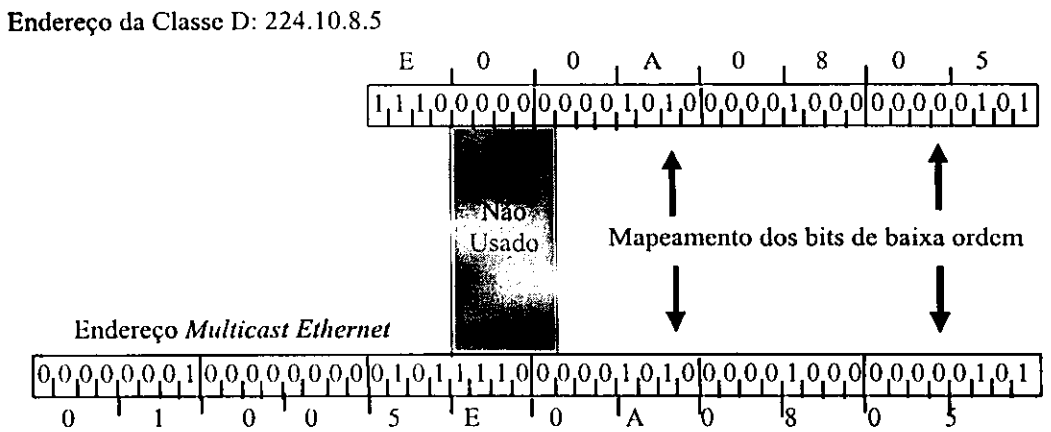


Figura 2. Mapeamento entre os endereços da classe D e endereços IEEE-802

O órgão responsável pela administração dos endereços é o IANA (*Internet Assigned Numbers Authority*). O endereço chave 224.0.0.0 não pode ser adotado por nenhum grupo, bem como a faixa de endereços 224.0.0.1 a 224.0.0.255, que é reservada para os protocolos de roteamento.

Os endereços restantes 224.0.1.0 a 239.255.255.255, podem ser utilizados pelas aplicações *multicast*, sendo que a faixa de endereços 239.0.0.0 a 239.255.255.255 é reservada para aplicações em um mesmo domínio (privado).

O *Multicast* em redes IP privadas (*Intranet*) assume a mesma definição do endereçamento na *Internet*, utilizando-se da classe D de endereçamento definida acima. Nas redes privadas podem existir grupos *multicast* formados somente por *hosts* internos à rede privada ou grupos formados por hosts que localizam-se fora e dentro da rede, caso essa tenha comunicação com outras redes externas.

Atualmente é muito comum que as redes privadas tenham comunicação com outras redes na *Internet* através de um *Firewall*[1] o que cria a necessidade da implementação de um túnel *multicast* que permita a passagem das informações através deste, ligando o roteador IP interno ao roteador IP externo.

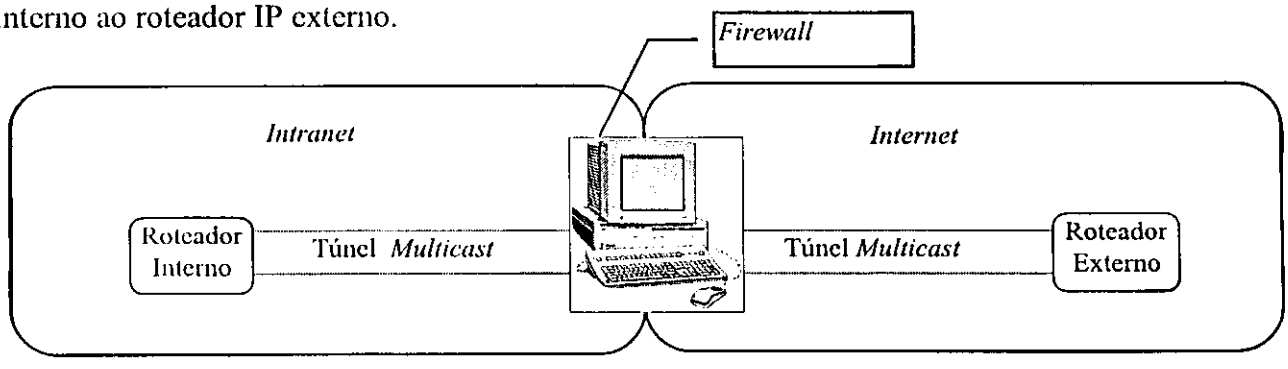


Figura 3. Túnel *multicast* entre uma *Intranet* e a *Internet*

Os grupos mais conhecidos de acordo com [SEMER 97] são:

- Todos os sistemas nesta sub-rede (224.0.0.1);
- Todos os roteadores nesta sub-rede (224.0.0.2);
- Todos os roteadores DVMRP (224.0.0.4);
- Todos os roteadores OSPF (224.0.0.5);
- Audio IETF (224.0.1.11);
- Vídeo IETF (224.0.1.12);
- AUDIONEWS (224.0.1.7); e
- MUSIC-SERVICE (224.0.1.16).

[1] Um *Firewall* é um computador configurado para proteger as redes contra invasões externas, limitando a passagem de informações através dele, de acordo com regras estabelecidas pelo gerente da rede.

2.1.2. Modelo de Gerenciamento de Grupos

É de responsabilidade dos *hosts* informarem aos roteadores sobre a sua participação em grupos *multicast*, possibilitando-os assim, contactarem outros roteadores, passando informações sobre as associações existentes, de forma que facilite o estabelecimento das rotas. A idéia é similar ao funcionamento dos roteadores convencionais.

Os *hosts* e roteadores que implementam *multicast* usam o IGMP [RFC 1112] para trocar informações à respeito dos grupos.

O IGMP é projetado para evitar congestionamento. A seguir veremos quais os cuidados que são tomados:

- Toda comunicação entre *hosts* e roteadores ocorre através de *multicast*, sendo as mensagens IGMP encapsuladas e transmitidas nos pacotes. Somente os *hosts* e roteadores *multicast* é que recebem as mensagens IGMP.

- Os *hosts* escutam respostas de outros *hosts* e suprimem algumas das suas respostas que são desnecessárias. Isso ocorre porque um roteador de uma rede local não precisa gravar quais os *hosts* que pertencem a um grupo, pois esta informação está na tabela de roteamento *multicast*. Há necessidade somente de saber se existe um *host* na rede local que pertença a um determinado par (Grupo, Fonte), não sendo necessário a manifestação dos demais *hosts* locais. Na prática, somente um *host* de cada rede ligado a cada par (Grupo, Fonte) responde uma mensagem solicitada por um roteador *multicast*.

- A taxa de solicitação à rede, em relação as alterações nos *hosts* associados a um par (Grupo, Fonte) é de, no máximo, uma por minuto.

Conceitualmente o IGMP possui três fases:

- A primeira fase realizada pelo IGMP [FENNE 97] é escolher o DR (*Designated Router*) ao qual vai ser atribuída a responsabilidade de desempenhar as funções do protocolo IGMP do lado dos roteadores.

A escolha ocorre após a troca de mensagens "*Hello*" entre os roteadores vizinhos. O emissor com maior endereço IP assume as funções de DR.

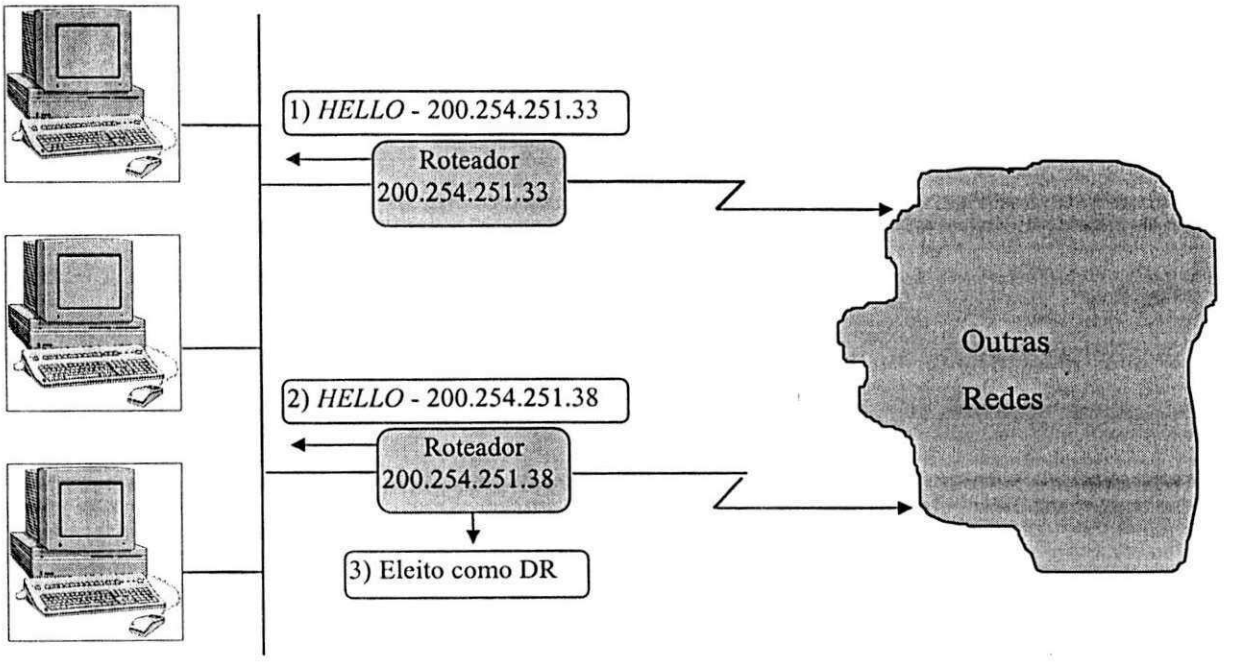


Figura 4. Escolha do *Designated Router*

• A segunda fase se caracteriza pela associação de um *host* a um par (Grupo, Fonte). Para esta associação é necessário que o *host* envie uma mensagem "*Inclusion Group Source Respond*" após o recebimento de uma mensagem "*Host Membership Query*" vinda do DR, indicando um par (Grupo, Fonte), como mostra a Figura 5.

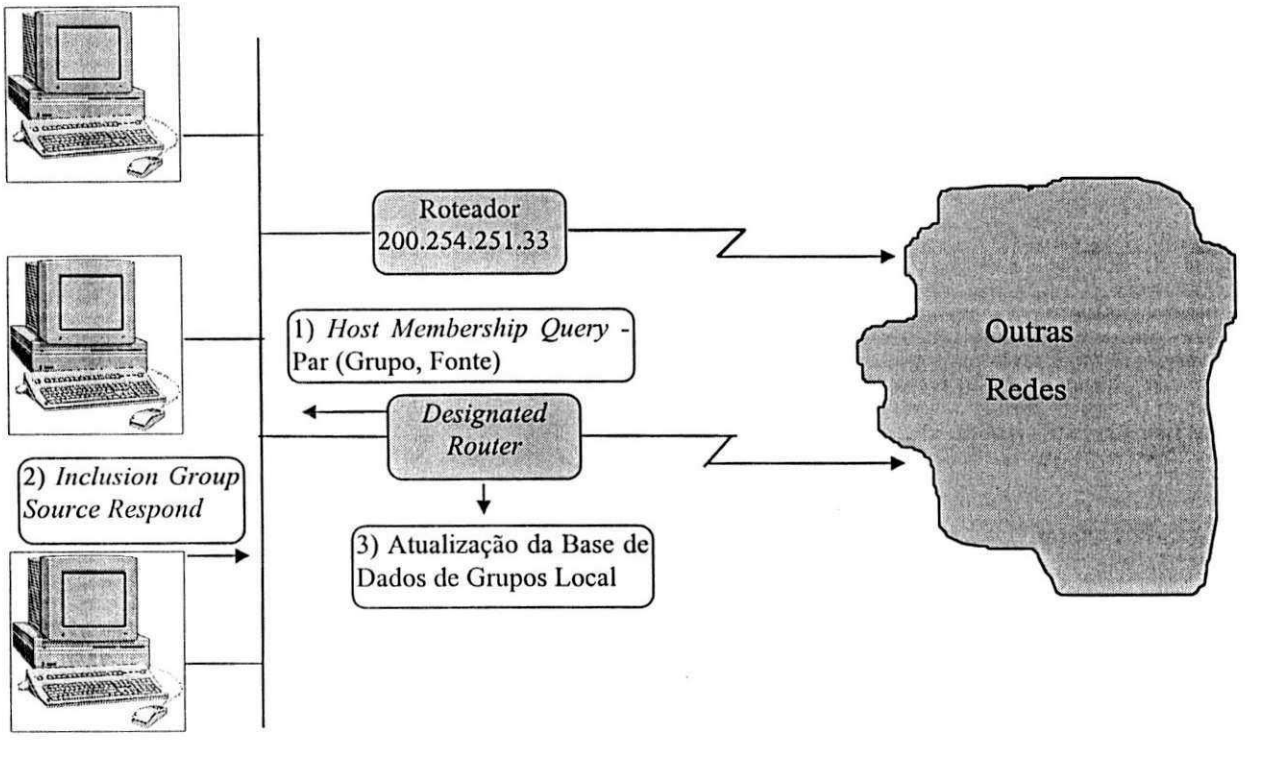


Figura 5. Associação a um par (Grupo, Fonte).

Após a troca das mensagem, é incrementado um apontador para o par (Grupo, Fonte) na "base de dados de grupos local", permitindo com isso que a rede receba as informações *multicast* destinadas ao par (Grupo, Fonte).

• A terceira fase é determinada pela fiscalização da permanência dos *hosts* às associações estabelecidas e o processo de desassociação a um par (Grupo, Fonte). Isso ocorre de tempos em tempos, através da geração da mensagem "*Host Membership Query*", por parte do DR, que pode ocasionar três comportamentos diferentes por parte dos *hosts* que fazem parte da rede local. São eles:

1) Uma nova associação, como vimos na segunda fase;

2) Uma desassociação explicita, através da *mensagem "Exclusion Group Source Respond"* gerada por um *host* da rede local, ocasionando um decremento no apontador do par (Grupo, Fonte) dentro da "base de dados de grupos local", como mostra a Figura 6; e,

3) Uma desassociação integral do par (Grupo, Fonte), indicado na mensagem "*Host Membership Query*", caso não ocorra resposta por parte dos *hosts* da rede.

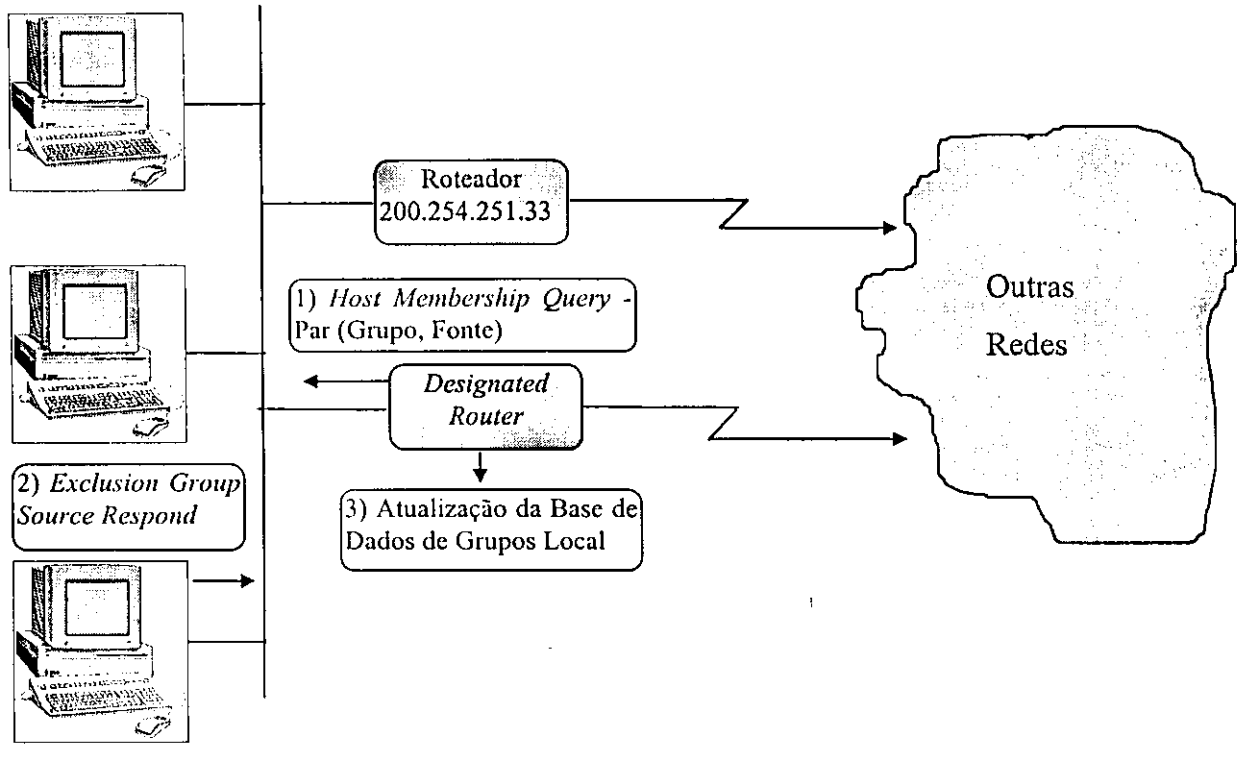


Figura 6. Desassociação de um *host* a um par (Grupo, Fonte)

Uma desassociação explícita pode ocasionar uma desassociação integral quando o contador para o par (Grupo, Fonte) em questão, ficar com seu valor igual a zero.

2.1.3. Modelo do Roteamento *Multicast*

Os roteadores *multicast* executam um protocolo para estabelecer os caminhos por onde um pacote deve seguir a fim de atingir todos os seus destinos. Alguns destes protocolos utilizam informações vindas de protocolos de roteamento *unicast*, como é o caso do DVMRP, que é uma extensão do RIP (*Routing Information Protocol*) [RFC 1058], ou MOSPF, que é uma extensão do OSPF (*Open Shortest Path First*) [RFC 1583]. Já outros, como é o caso do PIM-DM, PIM-SM e CBT, são independentes dos protocolos de roteamento *unicast*.

Todos esses protocolos, para encontrar os caminhos por onde devem passar os pacotes, são implementados a partir de algoritmos de roteamento, que veremos na seção seguinte.

2.2. Algoritmos de Roteamento *Multicast*

O IGMP provê somente a etapa final do serviço de entrega *multicast*, pois faz a entrega de pacotes aos membros do par (Grupo, Fonte) que estão diretamente ligados a um roteador. Quando for preciso a entrega de pacotes entre roteadores vizinhos ou através de uma rede faz-se necessário a participação de um protocolo de roteamento *multicast*.

O protocolo de roteamento é responsável pela construção de árvores de entrega *multicast* e pelo envio de pacotes através da árvore definida. Essa seção explora alguns algoritmos que podem, potencialmente, ser usados por protocolos de roteamento *multicast* para montar a árvore de entrega.

2.2.1. Flooding

Neste algoritmo, o roteador ao receber um pacote endereçado a um par (Grupo, Fonte), observa se ele não foi recebido anteriormente. Caso o pacote já tenha sido visto será

descartado. Caso contrário, o roteador o transmite em todas as interfaces, exceto a de chegada.

Para fazer o controle dos pacotes recebidos é inserido um número seqüencial de identificação em cada pacote gerado pela origem, desta forma, cada roteador deve manter uma lista de pacotes recebidos por origem. Para evitar que a lista cresça sem limites, cada uma das listas terá um contador que será incrementado até um valor máximo, determinando o tamanho da lista. Assim, após a lista estar cheia, o próximo pacote recebido deve entrar no lugar do primeiro pacote identificado na lista, como a estrutura de um fila (primeiro a entrar, primeiro a sair).

Este algoritmo é fácil de ser implementado, pois não precisa manter uma tabela de roteamento. Mas, por outro lado, é inapropriado para redes grandes, pois gera um número grande de cópias de pacotes e utiliza todos os caminhos presentes na rede. Além disso, exige muita memória dos roteadores para armazenar uma tabela com informações a respeito dos últimos pacotes recebidos [HARR 95].

2.2.2. SPANNING TREE

O algoritmo *Spanning Tree* é mais eficiente do que o *Flooding* porque escolhe apenas um caminho entre qualquer par de roteadores. Uma vez construída a árvore, o roteador simplesmente transmite cada pacote recebido através de todas as interfaces pertencentes à árvore gerada, exceto a de chegada.

Os problemas associados a esse algoritmo é a concentração de tráfego em poucos enlaces, e a utilização de caminhos que na maioria das vezes não são os mais eficientes entre a sub-rede de origem e os membros do grupo, devido à utilização dos mesmos caminho para todos os grupos existentes [IM 95].

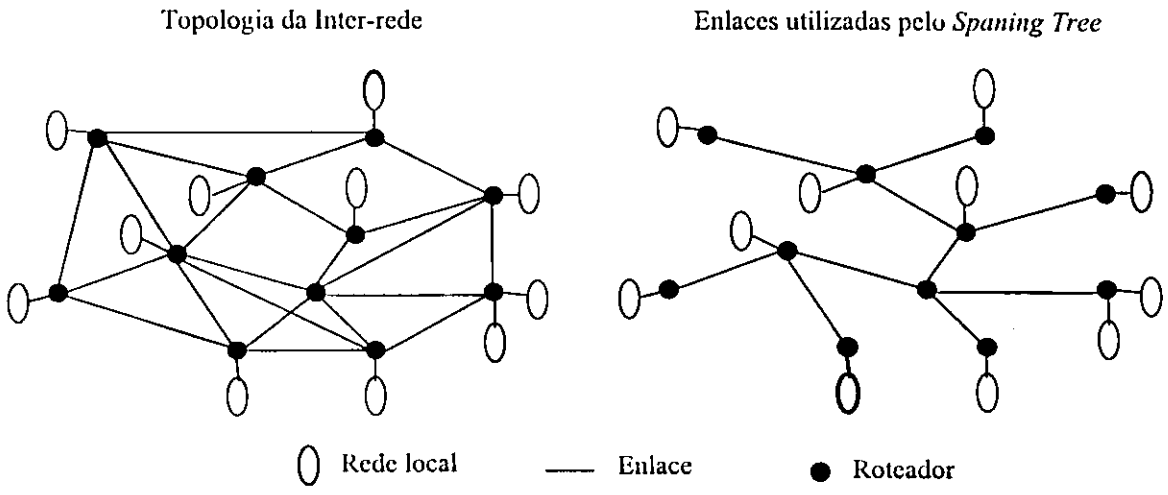


Figura 7. Demonstração das rotas utilizadas pelo *Spanning Tree*

2.2.3. REVERSE PATH BROADCASTING (RPB)

Ao invés de montar uma única árvore para todas as transmissões *multicast*, o RPB monta uma árvore para cada par ativo (Grupo, Fonte). Esta árvore é montada pelo RPB da seguinte forma: para cada par (Grupo, Fonte), se um pacote chegar na interface considerada pelo roteador como sendo a rota mais curta até a origem do pacote (chamado enlace-pai), este roteador transmitirá o pacote através de todas as demais interfaces (chamadas enlaces-filhos). Se o pacote chegar através de qualquer outra interface, ele será descartado.

Para reduzir duplicações desnecessárias de pacotes, o algoritmo pode ser estendido para determinar se o roteador vizinho num enlace-filho considera que o roteador local esteja no seu enlace-pai para este par (Grupo, Fonte). Se não estiver, então será suprimida a transmissão neste enlace, para evitar o descarte dos pacotes pelo vizinho. Essa informação sobre os enlaces do vizinho é fácil de determinar se estiver sendo usado um protocolo da classe *Link State*, porque neste caso cada roteador possui uma base de dados topológica para todo o domínio de roteamento. Ao usar um protocolo de roteamento da classe *Distance Vector*, o vizinho terá que anunciar qual é seu enlace-pai para cada (Grupo, Fonte) através do protocolo.

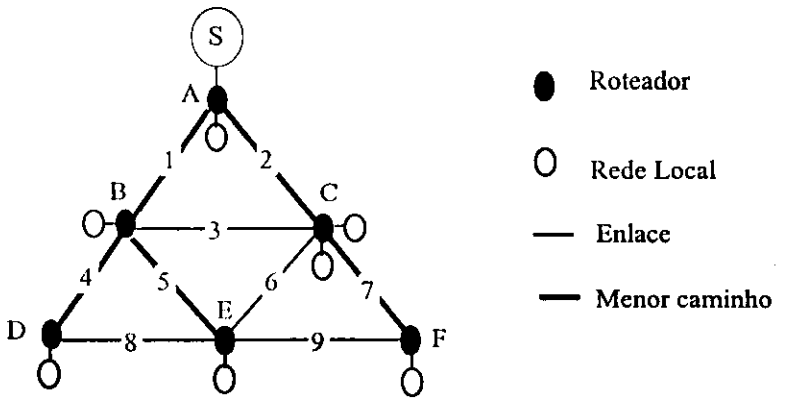


Figura 8. Exemplo do RPB

A Figura 8 mostra um exemplo da operação básica implementada pelo algoritmo RPB. Note que a fonte (S) está associada a uma rede diretamente conectada ao roteador A. Nesse nosso exemplo, vamos observar o roteador B, que recebe pacotes *multicast* do roteador A, considerando o enlace 1 como sendo seu enlace-pai para o par (Grupo, Fonte) e envia os pacotes pelos enlaces 4 e 5. Ele não os envia pelo enlace 3, pois sabe através do protocolo de roteamento que o enlace 2 é o enlace-pai do roteador C. Caso envie pacotes pelo enlace 3, estes vão ser descartados pelo roteador C.

O RPB é fácil de implementar e sempre gera a melhor rota ao grupo destino. A utilização da rede é mais eficiente, porque é calculada uma árvore distinta para cada par (Grupo, Fonte). Sua principal limitação é que ele encaminha pacotes até sub-redes onde não há membros do grupo destino [LEE 97].

2.2.4. TRUNCATED REVERSE PATH BROADCASTING (TRPB)

Este algoritmo dá uma solução para o problema apresentado pelo algoritmo anterior. Ele usa o IGMP para manter-se informado de quais sub-redes possuem membros de um determinado par (Grupo, Fonte), desta forma, só faz a distribuição de pacotes em sub-redes que apresentam membros associados à fonte [LEE 97].

2.2.5. REVERSE PATH MULTICAST (RPM)

O RPM cria uma árvore que inclui apenas roteadores e sub-redes ao longo do caminho mais curto até os membros do par (Grupo, Fonte) [STANT 96].

descartando a informação de "poda" dos roteadores.

Este algoritmo, embora melhor do que os anteriores, ainda sofre algumas limitações. Se quisermos estender o uso de *multicast* à rede IP inteira: primeiro, pacotes *multicast* precisam ser enviados periodicamente a todos os roteadores da rede; segundo, cada roteador precisa guardar o estado para seus pares (Grupo, Fonte). Estas limitações se tornam mais importantes, à medida que aumenta o número de fontes e grupos.

2.2.6. CORE BASED TREES (CBT)

Para diminuir a complexidade de manter-se uma árvore para cada par (Grupo, Fonte), o CBT utiliza uma única árvore de distribuição por grupo. O tráfego *multicast* é enviado e recebido pela mesma árvore independente da origem.

O núcleo da árvore consiste em um ou mais roteadores que formam a espinha dorsal do fluxo. A partir dos roteadores do núcleo é construída uma árvore para o grupo. Para passar a fazer parte do grupo, um roteador deve enviar uma mensagem de adesão a um roteador do núcleo, usando *unicast*. O pedido de adesão será processado por cada roteador onde passa, e marcará a interface de chegada como fazendo parte da árvore de distribuição do grupo. Esses roteadores intermediários continuam a retransmitir o pedido de adesão até que este chegue a um roteador do núcleo.

Um pacote enviado para o grupo é transmitido na forma *unicast*, destinado a um roteador do núcleo. Assim que alcançar um roteador pertencente à árvore, ele vai ser retransmitido em modo *multicast*.

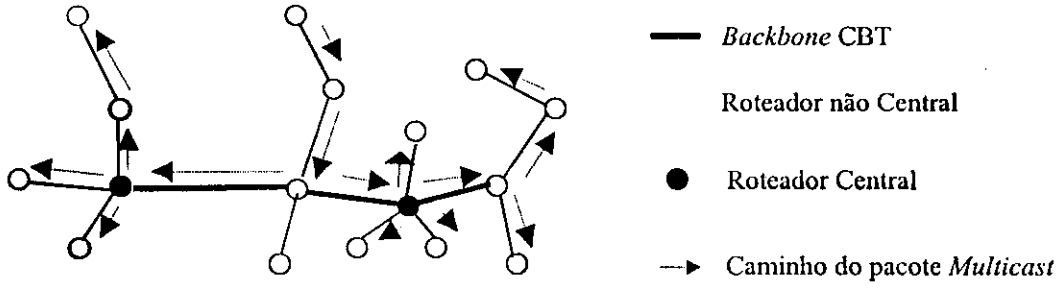


Figura 10. Árvore de Entrega *Multicast* CBT

O CBT é mais econômico em relação ao uso de recursos da rede do que o RPM, pois

só requer que os roteadores mantenham o estado para cada grupo, e não para cada par (Grupo, Fonte). Adicionalmente, evita a necessidade da retransmissão periódica de pacotes a todos os roteadores da rede. Suas limitações incluem a possibilidade de rotas com maior custo, aumentando o tempo de entrega, o que pode ser crítico para aplicações multimídia [SEMER 97].

2.3. PROTOCOLOS PARA PROVER QUALIDADE DE SERVIÇO

Atualmente, é necessário que a arquitetura TCP/IP possa prover serviços integrados, onde várias classes de tráfego compartilham a mesma banda passante com QoS (Qualidade de Serviço) diferentes, viabilizando assim, a crescente necessidade de serviços em tempo real para novas aplicações, incluindo videoconferência, seminários remotos e simulação distribuída, dentre outras.

Como espera-se que estas aplicações sejam providas através de transmissões *multicast*, neste ponto descrevemos a definição de QoS e os protocolos envolvidos, para que possamos identificar as necessidades impostas às transmissões *multicast* de acordo com as classes de tráfegos envolvidas.

No contexto de serviços integrados, QoS se refere à natureza do serviço que a transmissão de pacotes deve receber. Isso inclui a descrição da banda a ser reservada, o atraso dos pacotes e taxas de perdas de pacotes, dentre outros parâmetros possíveis [BRAND96].

De acordo com [BRAND96], os componentes básicos que devem estar presentes na arquitetura das redes que irão prover diferentes tipos de QoS são:

- **Especificação do Fluxo:** O usuário deve comunicar à rede a característica do fluxo de dados a ser gerado, e esta por sua vez, poderá identificar a QoS necessária a este fluxo.

- **Controle de Tráfego:** Como os recursos da rede são finitos, esta não pode atender a todos os pedidos de reserva de recursos. A arquitetura da rede precisa conter um conjunto de regras para aceitação das reservas, identificação dos fluxos de dados e controle de quando e como transmitir os pacotes.

- **Roteamento:** A rede precisa decidir como transmitir os pacotes da origem até o

destino. O roteamento deve ser capaz de definir os caminhos para comunicação *unicast* e *multicast*.

• **Reserva de Recursos**: A rede deve ser capaz de reservar os recursos necessários ao longo do caminho a ser percorrido pelo fluxo de dados.

O ponto central do comprometimento com a QoS está relacionado com o atraso dos pacotes levando-se em conta as aplicações em **Tempo Real**, as quais necessitam que os pacotes sejam entregues em tempo hábil, pois dados que chegam atrasados perdem a sua finalidade. Por outro lado, aplicações com **Tempo Elástico** sempre esperam a chegada de pacotes, isso não quer dizer que estas aplicações sejam insensíveis ao tempo, pelo contrário, aumentando o tempo de espera para a chegada de pacotes a performance das aplicações degrada-se significativamente [RFC 1301].

O compartilhamento de recursos está relacionado com o comprometimento que a rede assumiu com uma aplicação no momento da negociação da QoS. Quando uma aplicação necessita de uma quantidade de banda passante, esta deve negociar o recurso de rede para ter permissão de transmissão. O recurso não será viabilizado, caso venha a atrapalhar as outras aplicações com conexão já estabelecida.

Para que uma aplicação possa obter a QoS desejada, é necessário que esta obtenha permissão junto ao controle de admissão (ou protocolo de reserva). Este controle estabelece qual o nível de QoS que uma aplicação pode obter ou, simplesmente, não permite o estabelecimento da conexão. O controle de admissão requer que os roteadores conheçam as demandas que estão correntemente sendo feitas. Os cálculos, para avaliação dos recursos disponíveis na rede, são realizados em conformidade com parâmetros de serviços requeridos anteriormente [FIRO 95].

Após o estabelecimento de uma conexão, há necessidade de uma "monitoração" para prevenir abusos dos recursos da rede, ou seja, um árbitro que fiscalize as aplicações não permitindo que estas ultrapassem os parâmetros de QoS negociados durante o estabelecimento da conexão [RFC 1458].

Para que as redes possam dispor destes requisitos, a família de protocolos TCP/IP está sendo ampliada com novos protocolos que permitam a implantação da noção de QoS

desejada pelas novas aplicações. Estes protocolos são: o RTP (*Real-Time Transport Protocol*) (seção 2.5.1), o RTCP (*Real-Time Control Protocol*) (seção 2.5.2) e finalmente o RSVP (*Resource Reservation Protocol*) (seção 2.5.3).

2.3.1. REAL-TIME TRANSPORT PROTOCOL (RTP)

O RTP [RFC1889][RFC1890] é um protocolo de transporte que provê o serviço de entrega fim-a-fim para suportar aplicações transmitindo dados em tempo real, assim como, vídeo e áudio interativo. Esse serviço inclui identificação do tipo do conteúdo existente no pacote, número seqüencial, monitoramento de entrega, dentre outros, como podemos ver no desenho do cabeçalho RTP.

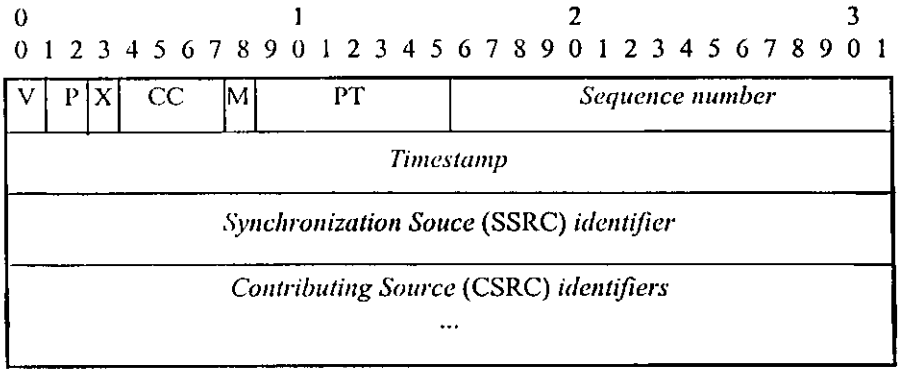


Figura 11. Campos do cabeçalho RTP

V: Identifica a versão do protocolo;

P: Identifica se o cabeçalho tem uma ou mais extensões;

X: Identifica se o cabeçalho tem a extensão pré-definida na sua normatização. Ver [RFC1889];

CC: Identifica que tipo de informações o pacote carrega;

M: Identifica um perfil para o pacote RTP;

PT: Identifica como a aplicação deve tratar os dados contidos no pacote, de acordo com o perfil designado no parâmetro M.

Sequence Number: Identificador da seqüência em que os pacotes são gerados pela origem;

Timestamp: Identifica o instante em que o pacote foi gerado;

SSRC: É usado para identificação de sincronismo entre origem e destino;

CSRC: É usado para identificar todas as fontes que fazem parte de uma sessão RTP.

O número seqüencial dos pacotes incluído pelo RTP simplesmente permite que o destino reconstrua a seqüência correta dos pacotes gerados pela fonte, embora possam faltar pacotes.

Através do *Timestamp*, o RTP permite a sincronização dos pacotes no destino através do controle de *jitter*[2] para viabilizar a exibição de áudio e vídeo.

O identificador do tipo de dados encapsulado no pacote RTP é importante para definir o tipo de compactação que este sofrerá, além de permitir a definição de diferentes sessões para diferentes tipos de mídia, permitindo com isso que o RTCP (seção 2.5.2) defina a qualidade de recepção para cada sessão. Por exemplo, áudio e vídeo são transmitidos em sessões distintas, permitindo ao receptor selecionar se deve ou não receber uma mídia específica em um determinado instante.

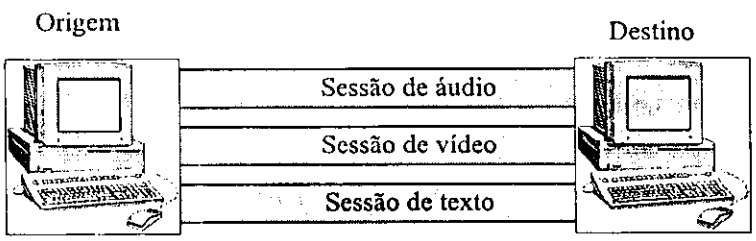


Figura 12. Estabelecimento de diferentes sessões RTP

As aplicações utilizam-se do RTP sobre o UDP para fazer uso de multiplexação e serviços de checagem, assim, ambos os protocolos contribuem para o funcionamento da camada de transporte. O RTP suporta transferência de dados para múltiplos destinos, desde que a rede esteja implementado com os protocolos de roteamento e gerência de grupos *multicast*.

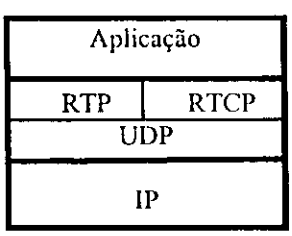


Figura 13.Localização do RTP e RTCP na pilha de protocolos

[2] *Jitter* é a variação do tempo de transferência fim-a-fim de pacotes (Atraso Máximo – Atraso Mínimo).

Embora o RTP inicialmente tenha sido desenvolvido para satisfazer as necessidades apresentadas pela videoconferência, ele pode ser estendido para outros tipos de aplicações como: transferência contínua de dados, simulação distribuída interativamente e controle e gerenciamento de aplicações. Atualmente o RTP está sendo usado no MBONE (seção 2.4.) e já existem aplicações comerciais desenvolvidas para várias plataformas.

2.3.2. REAL-TIME CONTROL PROTOCOL (RTCP)

O RTCP [RFC1889] [RFC1890] é o protocolo de controle que trabalha junto com o RTP, a fim de gerenciar os pacotes que são transmitidos periodicamente em cada uma das sessões do RTP para todos os outros participantes. A realimentação de informações para a aplicação pode ser usada para controlar o desempenho e diagnosticar problemas.

O RTCP executa quatro funções:

- **Provê informações para a aplicação:**

Esta função provê informações relacionadas à qualidade de distribuição dos dados. Cada pacote RTCP alimenta estatísticas úteis para a aplicação, definindo o número de pacotes enviados, número de pacotes perdidos, *jitter*, etc. Estas informações serão úteis para o remetente que pode modificar a sua forma de transmissão, ao receptor que pode determinar se existem problemas locais, regionais ou globais, e ao gerente da rede que pode avaliar o desempenho da distribuição *multicast*.

- **Identifica a fonte RTP:**

O RTCP carrega um identificador a nível de transporte da fonte RTP, chamado de "nome canônico" (CNAME), que é usado para obter informações dos participantes de uma sessão RTP. Os receptores usam o CNAME para associar os dados que fluem de um determinado participante, dentro de uma sessão para, por exemplo, sincronizar áudio e vídeo.

- **Controla os intervalos de transmissão:**

Para prover controle de tráfego e permitir que o RTP defina seções com o maior número de participantes possível, é definido um limite de 5% do tráfego total da sessão a

cada um deles. Este limite é calculado de acordo com a taxa de transmissão dos pacotes RTCP em função do número de participantes na sessão.

- **Carrega o mínimo de informações de controle da sessão:**

Como uma função opcional, o RTCP pode ser usado como um método conveniente para carregar informações como *login*, nome completo, e-mail, entre outros, a todos os participantes de uma sessão. Por exemplo, o RTCP poderia levar o nome pessoal para identificar um participante na exibição do usuário.

2.3.3. Resource Reservation Protocol (RSVP)

O RSVP é um protocolo de reserva de recursos desenvolvido para redes que suportam serviços integrados, permitindo que as aplicações requeiram uma determinada QoS fim-a-fim para a transferência de dados, independente da utilização de protocolos de roteamento *multicast* ou *unicast* [RFC 2210].

Em cada nó, o RSVP passa uma solicitação de reserva a uma rotina de Controle de Admissão, para verificar se há recursos suficientes disponíveis. Se existir, o nó reserva esses recursos no mecanismo de Controle de Tráfego. Uma vez feita a reserva é necessário que o RSVP também indique o filtro de identificação do fluxo de dados que fará uso da reserva. O Classificador de Pacotes do nó fará a identificação dos pacotes pertencentes ao fluxo que deve receber a reserva, verificará a rota a ser seguida e repassará estes pacotes ao Escalonador de Pacotes. Este então, tomará as decisões a respeito da transmissão dos pacotes, para obter a QoS solicitada. A Figura 14 ilustra a arquitetura do RSVP em um *host*/roteador [BRAND96].

Em cada nó, o RSVP se comunica com dois módulos de decisão locais, o Controle de Admissão e o Controle de Policiamento. O controle de Admissão verifica se há recursos disponíveis para a solicitação. O Controle de Policiamento determina se o tráfego está preservando o contrato com o nó e se o usuário tem permissão para fazer reservas. Se ambas as pesquisas obtiverem sucesso, o RSVP envia ao Controle de Tráfego as informações necessárias para que o Classificador de Pacotes e o Escalonador de Pacotes possam executar suas atividades. Se a verificação falhar, o RSVP retorna uma notificação de erro à aplicação que gerou a solicitação.

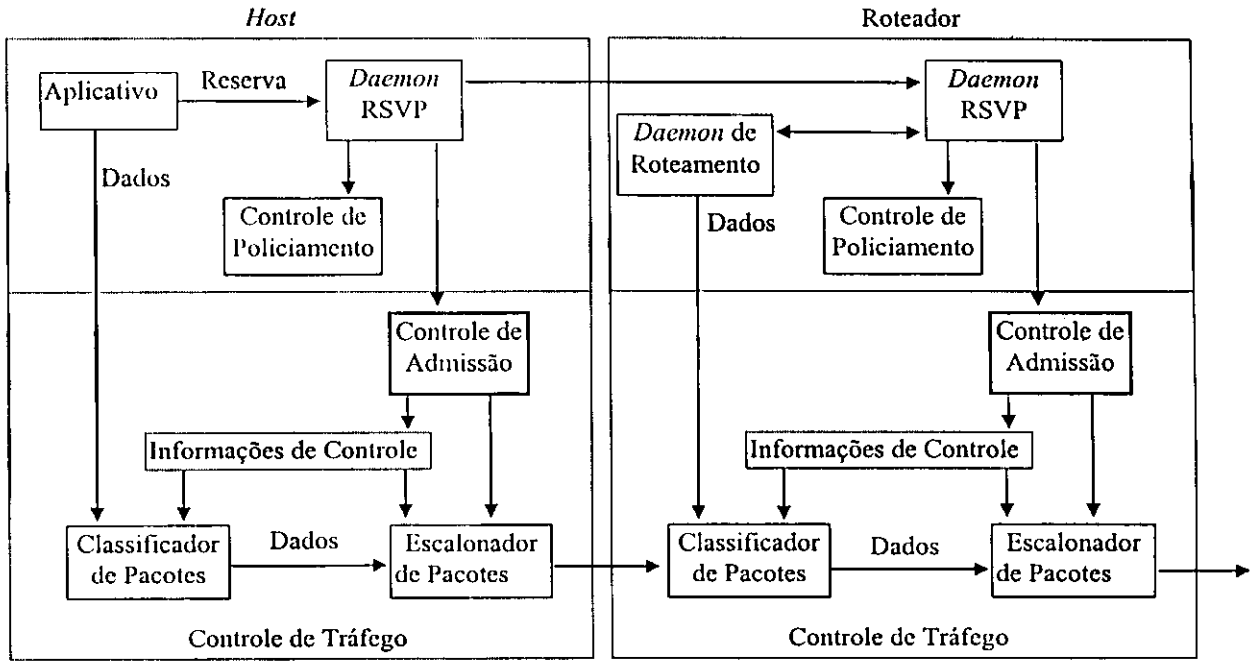


Figura 14. Arquitetura do RSVP

O RSVP opera sobre o IP (IPv4 ou IPv6), trabalhando em conjunto com um protocolo de transporte, porém não transporta nenhum dado. Ele é um protocolo de controle, assim como o IGMP ou ICMP (*Internet Control Message Protocol*) [RFC 0792], e usa os protocolos de roteamento para saber em que caminhos foram solicitadas as reservas, assim, quando o caminho de roteamento é mudado para um determinado tráfego que possui QoS garantida, ele deve tentar oferecer a mesma QoS, ou uma que continue satisfazendo a aplicação, caso contrário, a aplicação será finalizada [RFC 2210].

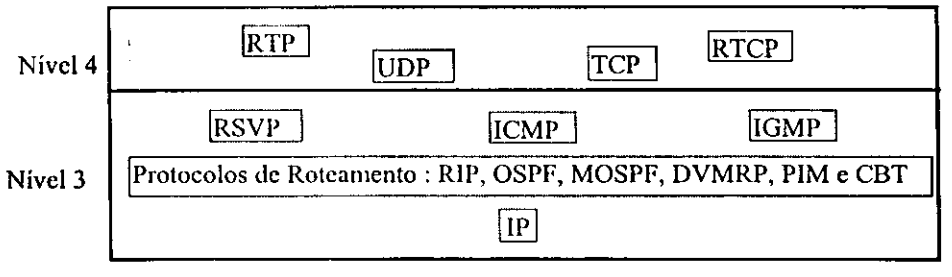


Figura 15. Pilha de protocolos da família TCP/IP

A reserva de recursos para *multicast* ocorre em paralelo à solicitação de associação de um *host* a um determinado grupo. Após o envio de uma mensagem IGMP, é enviada uma mensagem RSVP para garantir recursos ao longo do caminho de entrega do grupo. É importante salientar que todos os *hosts*, roteadores e outros elementos da infra-

estrutura das redes entre receptores e emissores devem suportar o RSVP.

A reserva de recursos deve ser independente para cada receptor, permitindo a alocação correta de recursos para cada um, caso contrário, ocorreriam problemas porque nem todos os membros de um grupo *multicast* possuem a mesma capacidade de processamento ou desejam a mesma QoS (como no caso da transmissão de vídeo para equipamentos com definições diferentes).

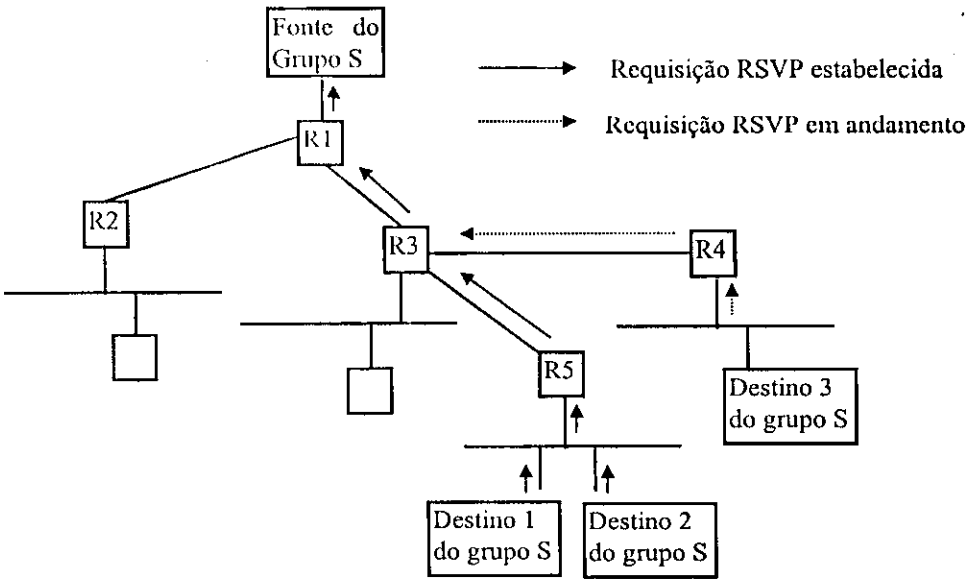


Figura 16. Requisição RSVP para recebimento de tráfego de uma fonte *multicast*

Outro problema do RSVP são os recursos computacionais requeridos pelos roteadores, no momento de selecionar e tratar os pacotes de acordo com suas prioridades. Desta forma, estão sendo desenvolvidos métodos baseados em etiquetas identificadoras para facilitar a identificação, e utilização de recursos de roteamento para prover caminhos alternativos e fixos [JOHNS 98].

O RTP complementa o RSVP, permitindo que as aplicações informem o desempenho da rede. Por exemplo, em aplicações multimídia, o áudio e o vídeo são transportados em sessões diferentes de RTP, separados em pacotes RTCP que controlam a qualidade da sessão; isso permite aos roteadores, com reserva de recursos estabelecidas, montar e gerênciar a reserva de banda passante.

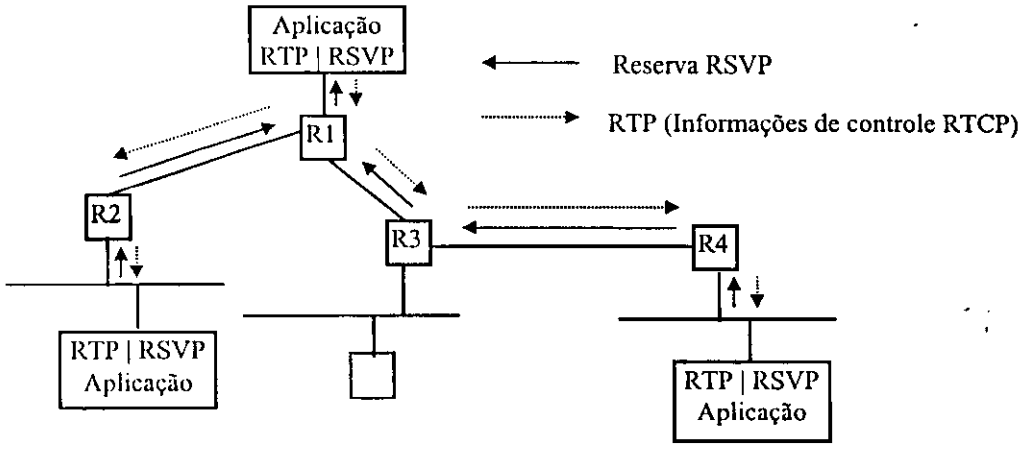


Figura 17.Uso de RSVP e RTP para uma aplicação multimídia sobre *multicast*

2.4. *BACKBONE MULTICAST* (MBONE)

A implantação *multicast* em toda a *Internet* deve demorar alguns anos, pois precisa da mobilização dos *sites*[3] existentes no sentido de implementar os protocolos *multicast*. Como esta decisão depende de cada gerente de rede e da necessidade de cada *site*, as implementações devem ser isoladas (em *Intranets*).

Para solucionar o problema de conectividade entre diferentes *Intranets* que estão implementadas com *multicast*, mas que não possuem vizinhos com as mesmas características, pretende-se estender o MBONE [CASNE 96] que é uma implementação de um *Backbone* virtual *Multicast* sobre a rede *Internet*.

Os objetivos do MBONE são:

- Promover a pesquisa para o desenvolvimento de protocolos *multicast*;

- Promover o desenvolvimento de novas aplicações multimídia que possam utilizar as facilidades *multicast*, e

[3] *Site* : Denominação dada a um determinado domínio. Um domínio neste contexto é um continuo conjunto de roteadores que são configurados para operar dentro de uma fronteira comum.

• Possibilitar a conectividade entre domínios diferentes, possibilitando desta forma a troca de informações entre regiões isoladas (ilhas *multicast*).

O MBONE utiliza roteadores *multicast*, que tipicamente são estações de trabalho que executam o *software mrouted*, o qual examina sua tabela de rotas para escolher quais interfaces devem ser utilizadas para transferir o pacote. As interfaces podem ser: redes locais implementadas com IP *multicast*, ou túneis que possibilitam a passagem dos pacotes *multicast* através de segmentos de rede que não usufruam de *software* que possibilite *multicast*.

Quando o *mrouted* necessita passar um pacote *multicast* através de um túnel, ele o coloca na área de dados de um pacote normal IP, com o endereço de destino apontando para o roteador que se encontra no outro lado do túnel. O roteador de destino recebe o pacote, o desencapsula e executa os mesmos procedimentos escritos acima. Falaremos mais sobre o funcionamento do *software mrouted* na seção 3.4, quando estivermos definindo o protocolo DVMRP no qual o *mrouted* é baseado.

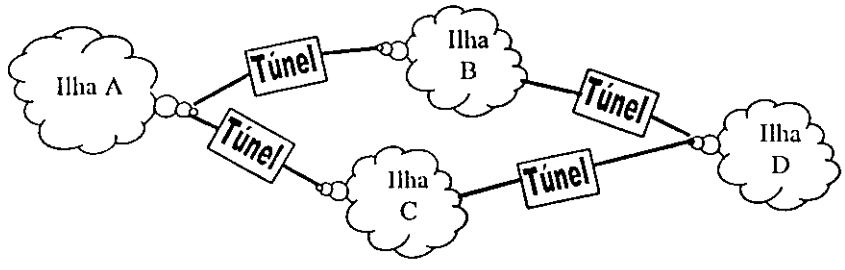


Figura 18. Conexão de ilhas *multicast* através de túneis

Atualmente, encontram-se em desenvolvimento experiências com outros protocolos de roteamento *multicast* como é o caso do MOSPF (seção 3.3), o PIM (seção 3.5) e o CBT (seção 3.6), possibilitando a ampliação e a interação entre os mesmos [HAWK 97].

A topologia é uma combinação de malha e estrela, sendo que a malha interliga o *backbone* a redes regionais, através do *mrouted* (DVMRP), enquanto que as estações ligadas as regiões formam uma topologia em estrela através do DVMRP, MOSPF, PIM e CBT.

A expectativa em relação ao futuro do MBONE é o rápido crescimento, com adesões em todo o mundo, provendo um meio que permita a passagem de tráfego com alta QoS, permitindo com isso que as aplicações multimídia possam ser utilizadas em toda a *Internet* [ESTR 97].

CAPÍTULO 3

PROTOCOLOS DE ROTEAMENTO *UNICAST* E *MULTICAST*

Os roteadores são os responsáveis pelo encaminhamento de informações (datagramas IP) entre as redes, viabilizando que uma informação gerada em uma rede possa chegar até *hosts* localizados em redes diferentes. Para executar essa atividade, os roteadores devem ter conhecimento da topologia da rede e das rotas utilizadas para encaminhar as informações ou simplesmente ter um caminho padrão definido pelo gerente da rede.

O conhecimento das rotas é armazenado em tabelas de roteamento, que podem ser mantidas de forma estática, com rotas definidas pelo gerente da rede, ou de forma dinâmica, com rotas mantidas através de protocolos específicos de manutenção de tabelas de rotas.

Os protocolos de roteamento tentam sempre manter a melhor rota entre dois pontos (no caso do *unicast*), que pode ser escolhida por um determinado custo (número de roteadores no caminho, capacidade dos enlaces, etc). No momento em que ocorre um problema em um enlace, automaticamente, o protocolo de roteamento ajusta a tabela de rotas para que a entrega das informações possa ser feita por outro caminho.

Existem atualmente dois protocolos mais disseminados para atualização dinâmica das rotas: o RIP (*Routing Information Protocol*) que é um protocolo de roteamento que utiliza o algoritmo *Vector-Distance* e o OSPF (*Open Shortest Path First*) que utiliza o algoritmo *Link-State*, ambos pertencentes a classe de protocolos IGP (*Interior Gateway Protocol*) [RFC 1371], que é responsável pela transferência de informações entre um mesmo Sistema Autônomo[4].

Os protocolos de roteamento *multicast*, em alguns casos, seguem as mesma filosofias dos protocolos de roteamento *unicast*, como é o caso do MOSPF que é uma extenção do OSPF e o DVMRP que é uma extenção do RIP. Ambos os protocolos MOSPF e DVMRP só podem ser implementados em conjunto com os protocolos *unicast* que os originaram.

Os outros protocolos *multicast* (PIM-DM, PIM-SM e CBT) foram desenvolvidos para trabalhar independentemente dos protocolos *unicast*, podendo conviver tanto com o OSPF como com o RIP, pois internamente foram desenvolvidos mecanismos que suprem as informações *unicast* (veremos neste capítulo).

Existem duas classificações para os protocolos *multicast*, os definidos para lidar com grupos densos, que é caracterizado por uma grande quantidade de estações e roteadores *multicast* no domínio da comunicação (DVMRP, MOSPF e PIM-DM) e, grupos esparsos, que são caracterizados por interligarem comunidades *multicast*, onde os participantes estão geograficamente espalhados (PIM-SM e CBT).

No restante deste capítulo veremos a especificação e operação dos protocolos *unicast* e *multicast* citados anteriormente, para que possamos obter embasamento teórico para o escopo que enfocará o restante deste trabalho. No final do capítulo apresentaremos um resumo comparativo das técnicas usadas para a implementação das árvores *multicast* e uma descrição rápida das vantagens e desvantagens de cada protocolo.

---

[4] Um Sistema Autônomo é um conjunto de redes e roteadores submetidos ao controle de uma única autoridade administrativa.

3.1. *ROUTING INFORMATION PROTOCOL* (RIP)

O RIP [MALK 97] [CARV 96] divide as máquinas da rede em ativas e passivas. As máquinas ativas divulgam informações de roteamento para as outras, enquanto as máquinas passivas recebem as informações e atualizam suas rotas, sem divulgá-las. Tipicamente, os roteadores executam o RIP no modo ativo, enquanto os *hosts* executam no modo passivo.

Um roteador executando o RIP no modo ativo difunde mensagens a cada 30 segundos ou, quando recebe uma informação de outro roteador, de acordo com a técnica *Poisson Reverse*. A mensagem difundida normalmente contém informações sobre toda a rede, extraídas da tabela de roteamento do roteador. Cada mensagem enviada por um roteador R consiste em pares de informações. Cada par é composto de um endereço de rede IP e da distância do roteador R à rede.

A métrica utilizada para o cálculo de distâncias é baseada no número de roteadores entre o roteador R e a sub-rede. O RIP assume o valor 1 para a distância de um roteador a uma sub-rede à qual ele está diretamente conectado. Para compensar diferenças de tecnologia de redes, algumas implementações do RIP informam uma distância maior quando a rota atravessa uma rede lenta.

As tabelas de rotas são mantidas e atualizadas pela troca das mesmas entre os roteadores que estão diretamente conectados. O roteador que recebe a tabela, a compara com a sua própria e modifica esta última nos seguintes casos:

- se o roteador emissor conhecer um caminho mais curto para uma determinada sub-rede, ou seja, se a distância apresentada na tabela do emissor for menor que a da tabela do receptor;

- se o roteador emissor apresentar uma sub-rede que o receptor não conhece, ou seja, se na tabela do emissor existir uma entrada que não está presente na tabela do receptor, esta entrada é inserida na tabela do receptor, e

- se uma rota que passa pelo emissor tiver sido modificado, ou seja, se a distância associada a uma sub-rede que passa pelo roteador emissor tiver mudado.

Para evitar situações de inconsistência entre roteadores, é utilizada a técnica de partição da rede chamada *Split Horizont*. Nesta técnica um roteador registra a interface da qual ele recebeu a informação de uma melhor rota e não propaga informações dessa rota sobre a interface. Tal técnica evita a seguinte situação:

☛ Suponha que o roteador R esteja diretamente conectado à rede N e informe esta distância ao roteador R'. Suponha que em um dado instante a conexão entre R e a rede N apresente problemas, tornando essa rede inacessível. Neste caso, R atualiza a rota para a rede N como inexistente. Se R receber de R' uma mensagem de roteamento antes de divulgar a queda da conexão com a rede N, R interpreta que a melhor rota para a rede N passa por R' e atualiza sua base de dados. Esta atualização causa a perda da informação da queda da conexão com a rede N e gera um ciclo no roteamento (R envia dados para a rede N via R', e R' envia dados para a rede N via R).

Outra técnica para contornar este tipo de problema é a retenção de informações. Esta técnica determina que um roteador, após receber a informação de que uma rede está inacessível, deve ignorar qualquer informação sobre a rede por um período fixo de tempo, assim permitindo que todas as máquinas da rede saibam da queda da conexão.

3.2. *OPEN SHORTEST-PATH-FIRST PROTOCOL* (OSPF)

O protocolo OSPF [RFC 1583] [CARV 96] é baseado nas mensagens: *Hello*, *Database Description*, *Link Status Request* e *Link Status Update*. Quando um roteador OSPF é inicializado, sua primeira ação é contactar os roteadores vizinhos, através de mensagens *Hello*. Os roteadores trocam mensagens entre si para eleger o DR (*Designated Router*). Este roteador torna-se responsável pela notificação de informações de roteamento a todos os roteadores presentes na rede (roteadores secundários).

A presença de um roteador mestre, com a função de gerar e distribuir informações, reduz significativamente o tráfego relativo às mensagens de roteamento, que são trocadas somente entre o roteadores mestres e os demais roteadores secundários.

As informações de roteamento trocadas entre roteadores, através da mensagem *Database Description*, indicam o estado e o custo associado às interfaces e aos

roteadores vizinhos. Estas mensagens são confirmadas pelos roteadores que a recebem.

A mensagem *Link Status Request* é usada por um roteador na requisição de dados atualizados a outro roteador, indicando quais enlaces são objetos de consulta. A resposta desta mensagem é através da mensagem *Link Status Update,* a qual contém as informações solicitadas sobre o estado dos enlaces em questão.

Uma vez estabelecido o roteador mestre de cada sub-rede, realizada a troca de informações de roteamento entre o roteador e os roteadores mestres das várias sub-redes em que esteja conectado, o roteador monta a sua base de dados de roteamento, obtendo como resultado uma árvore de roteamento, onde está posicionado na raiz, indicando a conectividade com outras redes. A partir dos dados de custo, são calculados os custos totais das rotas até cada sub-rede.

As facilidades listadas a seguir permitem ao OSPF diminuir a sobrecarga necessária para manutenção da topologia de um Sistema Autônomo:

- roteamento levando em consideração o tipo de serviço;
- balanceamento de carga entre rotas de mesmo custo;
- participação dos roteadores e redes em subgrupos denominados áreas, sendo a topologia de uma área conhecida apenas dentro da mesma, facilitando o crescimento modular de um sistema autônomo;
- definição de rotas específicas para máquinas e redes;
- designação de um roteador como representante de um grupo de roteadores para envio e recebimento de mensagens, de modo a minimizar o número de mensagens difundidas, e
- divulgação de informações recebidas de roteadores externos ao sistema autônomo.

O formato das mensagens OSPF permite distinguir informações recebidas de fontes externas daquelas recebidas dentro do Sistema Autônomo.

Com as vantagens listadas acima, podemos perceber que o MOSPF, que é uma

extensão do OSPF, tem todos os pré-requisitos para ser um eficiente protocolo de roteamento *multicast*, como veremos na próxima seção.

3.3. *MULTICAST EXTENSIONS OPEN SHORTEST PATH FIRST* (MOSPF)

Os roteadores implementados com MOSPF possuem uma imagem da topologia da rede, de acordo com as designações do protocolo de roteamento OSPF, especialmente construído para distribuir informações da topologia da rede entre roteadores de um mesmo sistema autônomo.

Os roteadores MOSPF utilizam-se do IGMP para monitorar as redes ligadas diretamente a eles, permitindo assim, que seja estabelecida uma "Base de Dados de Grupos Local" (BDGL), onde são mantidos os membros de um determinado grupo e especificado o roteador local que será responsável pela entrega dos pacotes *multicast* a esses membros.

Para que os roteadores possam manter a BDGL, é escolhido um roteador MOSPF, chamado *Designated Router* (DR), que envia mensagens IGMP à rede, perguntando se existem membros de um determinado par (Grupo, Fonte) e espera por mensagens IGMP de resposta, as quais servem para atualização da "Base de Dados de Grupos Local".

O DR é responsável ainda, pela transferência via *Flooding* das informações existentes na BDGL aos outros roteadores no domínio OSPF, criando desta forma, a *Group-Membership Advertise Link-State*, assegurando com isso, que todos os pacotes originados remotamente possam ser transmitidos a outros membros de um determinado par (Grupo, Fonte). Isso é possível pela criação da árvore SPT (*Shortest Path Tree*) entre uma fonte de um grupo e seus demais membros.

Cada SPT é construída sob demanda, quando o primeiro pacote atinge um membro do par (Grupo, Fonte) destino, utilizando os mesmos procedimentos do protocolo OSPF (detalhado na seção anterior), a fim de construir a árvore no sentido do roteador fonte para o roteador destino. Após a construção da árvore, inicia-se o processo de "poda" dos galhos que não são necessários para a transferência dos pacotes.

Vamos considerar a Figura 19 para podermos ter uma visão ampla do resultado da criação de uma árvore SPT, originada de uma Fonte S para um Grupo G. Além disso, vamos utilizar o "Roteador E" para indicar como são denominados os posicionamentos dos roteadores na árvore. Por exemplo, o "Roteador B" é chamado de nodo *upstream* e as "sub-redes 6 e 7", de *downstream*, em relação ao "Roteador E". Com isso podemos sugerir que o "Roteador E" não possui membros do Grupo G para a Fonte S, mas faz parte da árvore *multicast*, devido ao fato de ser caminho para outros roteadores que possuem membros do Grupo G para a Fonte S.

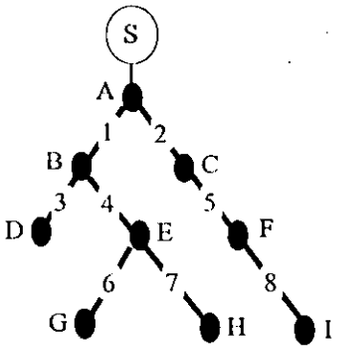


Figura 19. Árvore SPT para um par (Grupo G, Fonte S)

Os roteadores MOSPF tomam as suas decisões de entrega de pacotes de acordo com informações armazenadas no *Forwarding Cache*, que é construída a partir da árvore SPT de um par (Grupo, Fonte) e da "Base de Dados de Grupos Local" de cada roteador. Para isso, o roteador descobre sua posição na árvore *multicast* e cria entradas no *Forwarding Cache* de acordo com os seus *upstream* e *downstream*. Após a criação das entradas, a árvore *multicast* é descartada, liberando os recursos associados com sua criação. Desse ponto em diante, os pacotes do par (Grupo, Fonte) são entregues de acordo com o *Forwarding Cache*.

Destino	Origem	*Upstream*	*Downstream*	*Time to Live* - TTL
224.1.1.1	128.1.0.2	1	2, 3	5
224.1.1.1	128.4.1.2	1	2, 3	2
224.1.1.1	128.5.2.2	1	2, 3	3
224.2.2.2	128.2.0.3	2	1	7

Figura 20. Exemplo do *Forwarding Cache* MOSPF

Na Figura 20 mostramos um exemplo do *Forwarding Cache* de um roteador MOSPF. Os elementos apresentados na figura são:

• Destino: É o endereço do grupo destino para onde o pacote vai ser enviado.

• Origem: É o endereço *unicast* do *host* que gerou o pacote. Cada par (Grupo, Fonte) deve ter uma entrada diferente no *Forwarding Cache.*

• *Upstream*: É a interface pela qual o pacote será recebido.

• *Downstream*: São as interfaces através das quais o pacote será enviado para chegar aos membros do par (Grupo, Fonte).

• TTL: É o número máximo de saltos que um pacote pode dar para alcançar os membros de um grupo, permitindo com isso, que os roteadores possam descartar os pacotes que ultrapassem este valor.

As informações armazenadas no *Forwarding Cache* não são renovadas ou atualizadas, a menos que exista uma troca na topologia, gerada pelo protocolo OSPF, ou haja advertências de mudanças nos membros de um grupo, indicado pelo envio de *Group-Membership Advertise Link-State.*

Roteadores OSPF e MOSPF podem coexistir em um mesmo domínio, o que permite uma implementação gradativa do protocolo MOSPF e possibilita experiências *multicast* numa escala reduzida.

Quando os membros de um grupo residem em áreas OSPF diferentes, a forma de entrega dos pacotes continua sendo determinada pelo conteúdo do *Forwarding Cache*, porém, existem diferenças relacionadas ao modo com que as informação de associação aos grupos são propagadas e como as árvores SPT são montadas entre as áreas.

Para ser viabilizada a transferência *multicast* entre várias áreas diferentes, há necessidade da troca de informações relacionadas aos grupos e da entrega dos pacotes *multicast*. Estas funções devem ser implementadas por roteadores denominados BR (*Border Router*), em todas as áreas que necessitam comunicação com outras.

Um conjunto de BRs de áreas diferentes, ao fazerem parte de um *Backbone*, possibilitam a entrega de tráfego *multicast* entre as áreas, a partir da introdução do conceito

de "Recebedor *multicast Wild-Card*", que são roteadores que recebem todos os pacotes gerados para os grupos, independente de fazerem ou não parte do grupo. Isso garante que o tráfego *multicast* gerado em uma área seja entregue ao *backbone*.

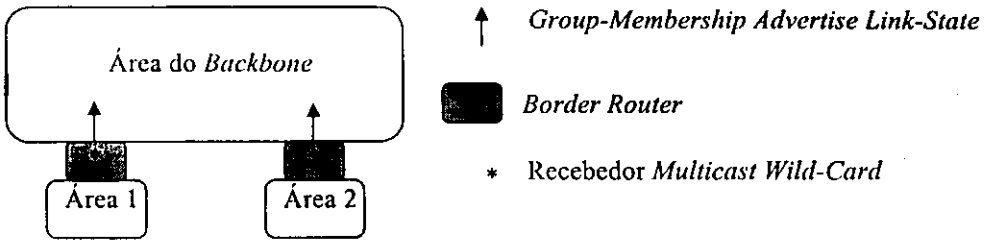


Figura 21. Arquitetura do Roteamento entre áreas

Existem dois casos que necessitam ser considerados, quando a árvore SPT é construída entre áreas diferentes:

• Se a fonte de um pacote *multicast* localiza-se na mesma área onde foram feitos os cálculos da árvore SPT, os galhos que levam aos roteadores implementados com *Wild-Card* não devem ser podados permitindo, assim, que as informações possam passar a outras áreas.

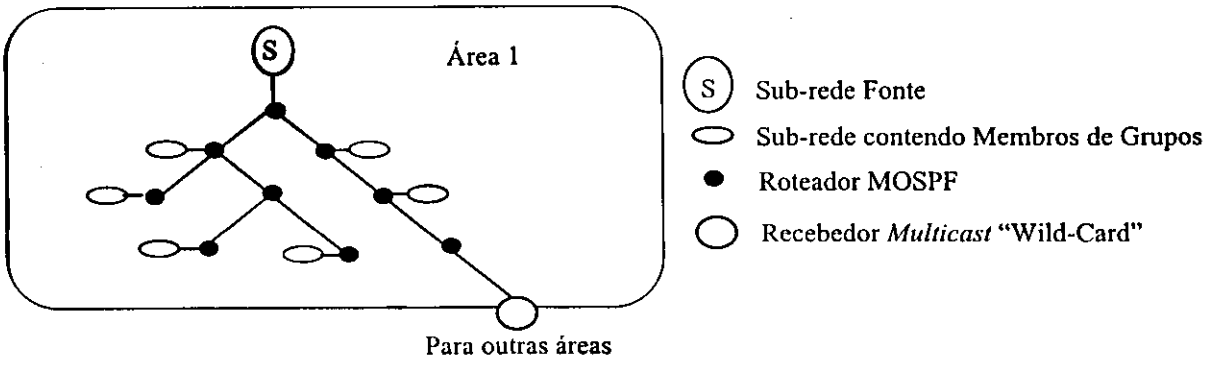


Figura 22. Fonte na mesma área dos cálculos da árvore SPT

• Se a fonte *multicast* residir em uma área diferente da qual o roteador faz o cálculo, os detalhes da topologia local não são conhecidos integralmente. Entretanto, as informações podem ser estimadas, através do uso das mensagens de *Link-State Advertise*, que são trocadas entre as áreas. Neste caso, na área de destino, a raiz da árvore começa no BR, como nos mostra a Figura 23.

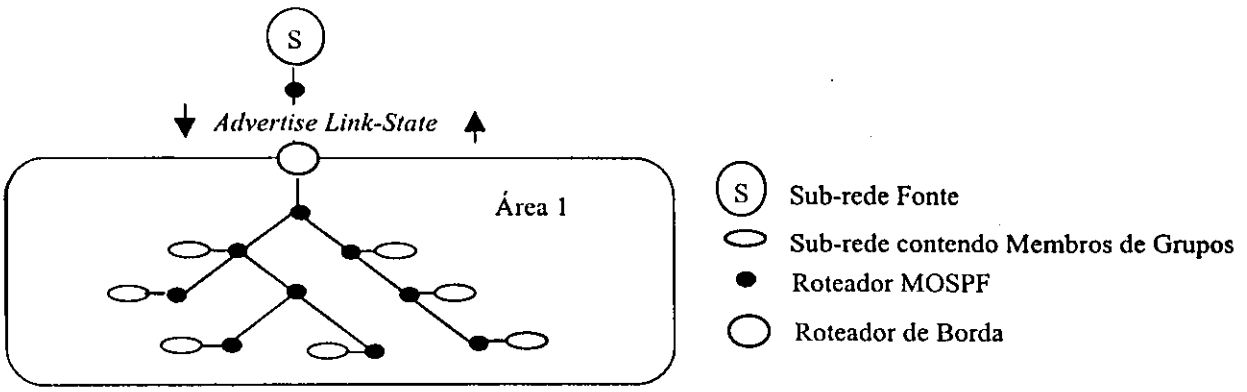


Figura 23. Fonte em área diferente da qual o roteador faz o cálculo da árvore SPT

A troca de informações *multicast*, entre áreas que implementam o protocolo MOSPF e áreas que implementam outros tipo de protocolos, assim como DVMRP, CBT e PIM, são definidas de forma semelhante a trocas de informações *multicast* entre áreas MOSPF. Para maiores detalhes, ver [SEMER 97].

3.4. *DISTANCE VECTOR MULTICAST ROUTING PROTOCOL* (DVMRP)

O DVMRP tornou-se popular após a implementação do *software mrouted*, utilizado em grande escala no MBONE (seção 2.4). O *mrouted* foi desenvolvido tomando como base o DVMRP, com a introdução de "podas" nos ramos inúteis das árvores inicialmente construídas para um par (Grupo, Fonte), de acordo com o algoritmo RPM (seção 2.2.5).

As portas de um roteador DVMRP podem ser uma interface física ligada a uma sub-rede local, ou uma interface ligada através de túneis até uma outra ilha de conectividade *multicast*. Um túnel é um enlace ponto a ponto virtual ligando dois roteadores *multicast*, por onde trafegam pacotes *multicast* encapsulados em pacotes *unicast*, como visto na página 27. Isto faz com que os roteadores que compõem este enlace tratem este tráfego como pacotes IP ponto-a-ponto. Nesse caso, os túneis são usados devido à existência de roteadores que não suportam *multicast*.

O roteador transmitirá um pacote através de uma interface *multicast*, desde que o valor do TTL (*Time To Live*) não exceda o limiar apropriado à interface, de acordo com a tabela abaixo. Por exemplo, um pacote com o TTL menor que 32 é restrito para o mesmo *site* e não será enviado por uma interface que o leve a outro na mesma região.

TTL Inicial	Restrição de escopo
1	mesmo *host*
16	mesma sub-rede
32	mesmo *site*
64	mesma região
128	mesmo continente
255	sem restrição

Tabela 1. Valores de controle do TTL

A descoberta dos roteadores que estão executando o protocolo DVMRP ocorre através da troca de mensagens chamadas *Neighbor-Probe*, permitindo que cada roteador guarde informações a respeito de quais roteadores em um domínio estão habilitados para recebimento de tráfego *multicast* DVMRP.

A recepção de informações, vindas de seus vizinhos, permite que um roteador reavalie as rotas alternativas para todos os destinos e adote as rotas de menor custo. Estas informações são trocadas através da técnica *Poisson Reverse*, que serve para atualização da lista de roteadores *downstream* e, descoberta do roteador *upstream*. Uma mudança de topologia se propaga lentamente nesta abordagem, sendo necessário o cálculo da tabela de rotas a cada roteador, antes de prosseguir a divulgação das rotas.

Inicialmente, é enviado o primeiro pacote de uma fonte a todos os nodos no escopo de abrangência permitido pelo TTL, fazendo com que os roteadores de menor custo até a fonte enviem os pacotes a todos os roteadores que fazem parte de sua lista de *downstream*.

Quando o pacote chegar aos roteadores folha[5], estes perguntam à sub-rede associada diretamente a eles se existe interesse em receber pacotes do par (Grupo, Fonte), utilizando mensagens IGMP.

[5] Roteadores folha são detectados por não receberem mensagens através da técnica *Poisson Reverse*.

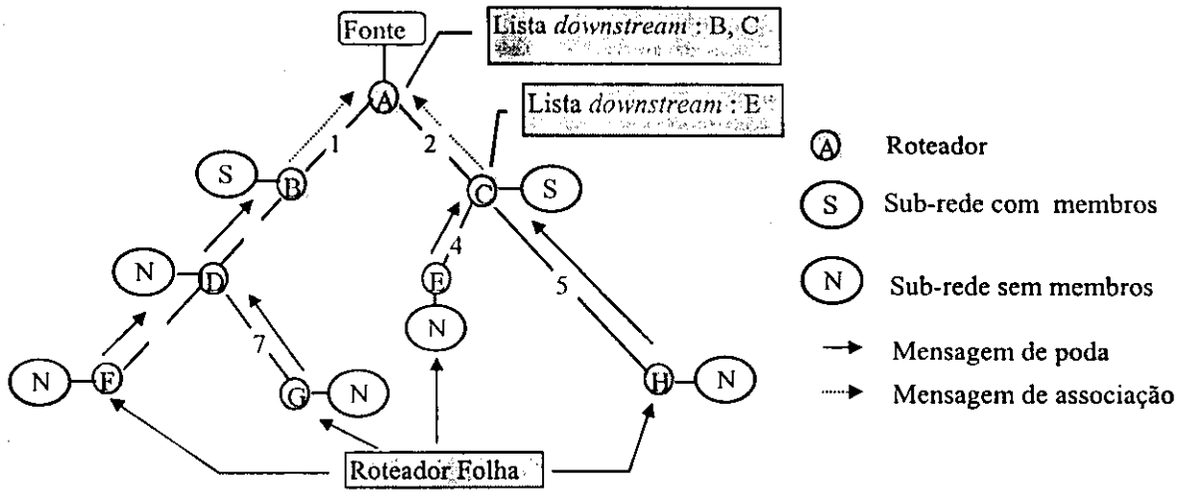


Figura 24.Construção da árvore *multicast* DVMRP

Caso um dos *hosts* da sub-rede local responda afirmativamente, é criada uma entrada na tabela de entrega DVMRP, entretanto, se nenhuma resposta dentro de um certo tempo for recebida pelo roteador, este envia uma mensagem de "poda" a seu *upstream*, indicando que não necessita receber pacotes do par (Grupo, Fonte) indicado.

A mensagem de "poda" vai causar a remoção de uma entrada na lista de *downstream*, eliminando a entrega de pacotes do par (Grupo, Fonte) para esta interface. Caso a lista de *downstream* fique vazia, o roteador envia uma mensagem de "poda" para o seu *upstream*, indicando que o enlace deve ser removido da árvore de distribuição *multicast*.

O estado de "poda" somente será alterado, se após uma outra solicitação IGMP do roteador à rede, esta for respondida, indicando que um novo *host* quer fazer parte do par (Grupo, Fonte), causando um "reenxerto" aos enlaces previamente podados.

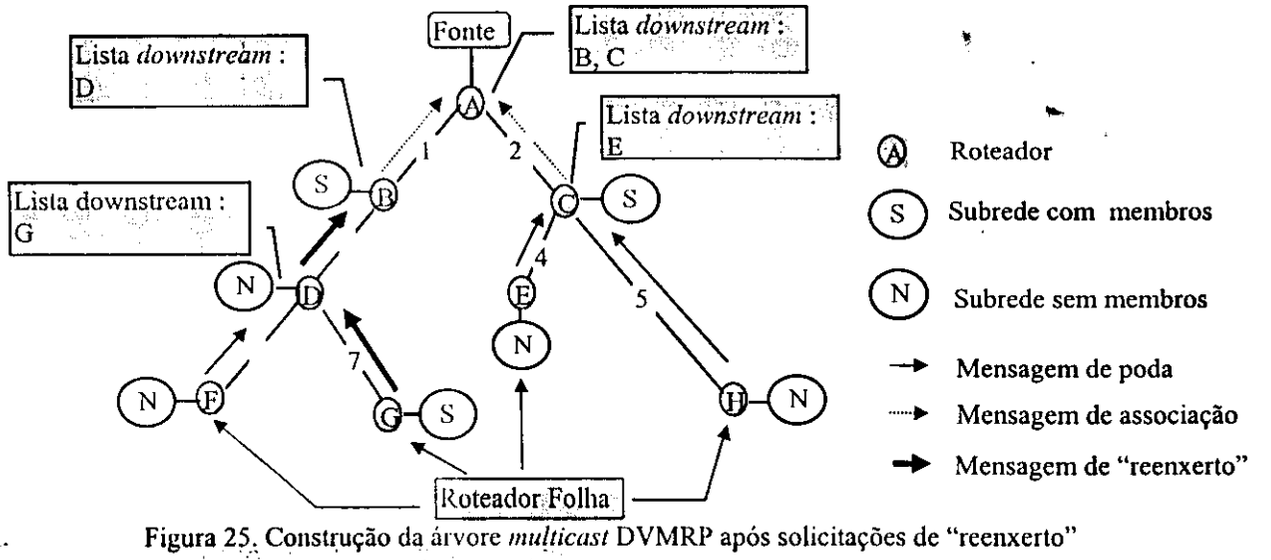


Figura 25. Construção da árvore *multicast* DVMRP após solicitações de "reenxerto"

Os problemas de escalabilidade, inerentes aos protocolos *Vector-Distance*, determinaram o início do desenvolvimento de um DVMRP hierárquico que não trata mais uma inter-rede como um único espaço de roteamento, dividindo-o em domínios, como ocorre nos demais protocolos de roteamento *multicast*.

O roteamento hierárquico reduz a ocupação de recursos dos roteadores, porque cada roteador precisa somente conhecer os detalhes sobre o roteamento de pacotes para seus destinos dentro de seu domínio. Com a redução das informações de roteamento que precisam ser armazenadas pelos roteadores, outros benefícios surgem principalmente para a utilização do DVMRP no MBONE:

- Diferentes protocolos *multicast* podem ser desenvolvidos em cada região do MBONE, facilitando o teste e o desenvolvimento de novos protocolos.

- Os efeitos de uma falha em um enlace se restringe apenas a um domínio, o que é importante para protocolos baseados no método *Vector-Distance*, devido ao longo tempo de atualização das tabelas em cada roteador.

Para a comunicação entre as regiões, é definido um roteador chamado BR (*Border Router*), responsável por passar os pacotes gerados de uma região para outra, como mostra a Figura 26 [THYA 96].

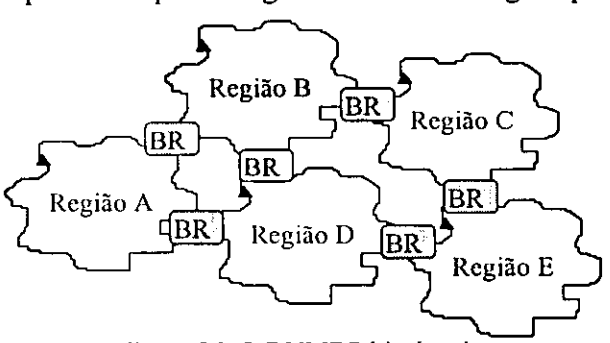


Figura 26. O DVMRP hierárquico

A forma de um *host* obter associação a um grupo com fonte em uma outra região é equivalente a associação a um par (Grupo, Fonte), entretanto, é definido como sendo um par (Grupo, Região). Para maiores informações a respeito do DVMRP, consultar [PUSAT 96].

3.5. *PROTOCOL INDEPENDENT MULTICAST* (PIM)

O PIM é um protocolo de roteamento *multicast*, em desenvolvimento, que utiliza informações sobre rotas ponto-a-ponto, obtidas através de um protocolo de roteamento *unicast* arbitrário. O PIM também distingue a forma de roteamento para grupos densos ou esparsos, como veremos nas próximas duas seções ao descrevermos o PIM-DM (*Protocol Independent Multicast - Dense Mode*) e o PIM-SM (*Protocol Independent Multicast - Sparse Mode*).

3.5.1. PROTOCOL INDEPENDENT MULTICAST - DENSE MODE (PIM-DM)

Embora a estrutura do PIM tenha sido desenvolvida inicialmente para prover escalabilidade em árvores de entrega *multicast* esparsamente distribuídas, ela também define um novo protocolo para o modo denso, chamado PIM-DM.

O PIM-DM assume que todas as interfaces *downstream* querem receber um pacote *multicast*, o que é considerado bom para grupos densos. Caso existam áreas de uma rede que não possuam membros de um determinado grupo, "podas" serão executadas nos galhos da árvore.

Para livrar-se da dependência dos protocolos *unicast* o PIM-DM suporta algumas duplicações de pacotes, até montar a estrutura definitiva de uma árvore para um par (Grupo, Fonte). Os três mecanismos básicos usados para construir as árvores *multicast* são: "poda", "reenxerto" e detecção de "redes folhas".

Um roteador sabe da existência de outros roteadores na mesma sub-rede através da troca de mensagens de "*Hello*". Caso um roteador não receba mensagens de "*Hello*", ele é o único roteador da sub-rede. É através deste método que é escolhido o DR (*Designated Router*) de cada rede, pois no pacote "*Hello*" é colocado o endereço de cada roteador, sendo eleito o de maior endereço IP.

Assim como no DVMRP, o algoritmo usado para montar as árvores de distribuição *multicast* é o RPM, entretanto, como inicialmente, são enviados pacotes em todas as interfaces, o recebimento dos mesmos em uma sub-rede deve ser solucionado através de

mensagens de "*Assert*", que define qual a interface que deve receber o pacote, de acordo com o menor custo até a fonte. Caso existam interfaces com o mesmo custo, a de maior endereço IP é que será escolhida.

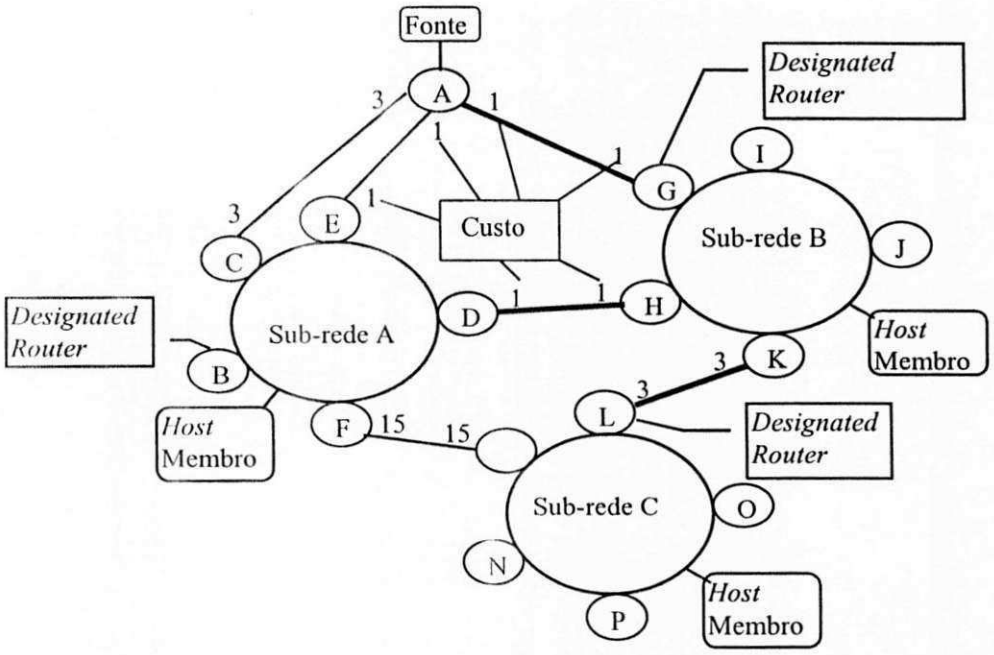


Figura 27. Definição dos enlaces utilizados para a construção da árvore *multicast*

A mensagem de "*Assert*" carrega o custo de cada roteador para atingir a fonte de um par (Grupo, Fonte) e o endereço IP do roteador que a originou. Esta mensagem é enviada para o endereço 224.0.0.13, para que todos os roteadores na sub-rede possam recebê-la e montar sua lista de interface de saída. Os roteadores que descobrirem que não possuem os melhores enlaces para receberem os pacotes da fonte, enviam mensagens de "poda" para seus *upstreams*, causando a remoção deste roteador da lista de interfaces de saída.

De acordo com a Figura 27, após o pacote inicial da fonte ter atingido as sub-redes por várias interfaces, as mensagens de "*Assert*" e subseqüentes "podas" definirão que o caminho (enlaces em negrito na figura) para atingir as sub-redes são :

sub-rede A: A → G → H → D

sub-rede B: A → G

sub-rede C: A → G → K → L

Após a árvore de entrega *multicast* ser montada, os *hosts* associam-se aos grupos ou desassociam-se através de mensagens IGMP e subseqüentes mensagens de "poda" e "reenxerto", como acontece no protocolo DVMRP.

O PIM-DM não define a troca de informações entre sistemas autônomos, utilizando-se do PIM-SM para prover este recurso. Maiores informações a respeito do PIM-DM podem ser encontradas em [HELMY 97].

3.5.2. PROTOCOL INDEPENDENT MULTICAST - SPARSE MODE (PIM-SM)

O DVMRP, MOSPF e PIM-DM são adequados quando usados em regiões onde os grupos estão amplamente representados, ou onde a banda passante é abundante. Entretanto, quando os membros dos grupos e suas fontes estão esparsamente distribuídos em grandes áreas, esses esquemas não são eficientes, pois os pacotes de dados ou informações aos membros são periodicamente enviados por inúmeros enlaces desnecessários.

Essa característica faz com que a comunidade TCP/IP investigue alternativas de roteamento *multicast* que estabeleçam árvores de distribuição eficientes através de grandes áreas em redes IP, onde muitos grupos estão esparsamente representados, e onde a banda passante não é uniformemente abundante.

Para um *host* juntar-se a um grupo, ele envia uma mensagem IGMP "*Inclusion Group Source Repond*" de resposta ao DR após receber uma mensagem "*Host-Membership-Query*", como mostrado na Figura 29, nas ações 1 e 2.

Quando existem vários roteadores conectados a uma rede, um deles será escolhido para operar como DR por um determinado tempo. O DR é responsável pelo envio de mensagens de "Registro" e "Junção/Poda" em direção ao RP (*Rendezvous Point*), além de implementar as funções do IGMP.

O PIM-SM usa a abordagem do CBT (*Core Based Trees*), no sentido de definir para cada grupo o conceito de RP [SEMER 97], onde receptores se encontram com novas fontes. O precursor de um grupo designa um RP primário e uma pequena lista ordenada de RPs alternativos.

A definição de qual roteador será o RP de um domínio, pode ser de duas formas, ou manual, configurado pelo gerente do domínio (forma estática), ou através do mecanismo "*Bootstrap*" (forma dinâmica).

O mecanismo "*Bootstrap*" define dentro do domínio um roteador chamado BSR (*Bootstrap Router*) para ser o responsável pela escolha de um RP, a partir de um conjunto de C-RPs (RPs candidatos).

Os C-RPs, periodicamente, enviam mensagens "C-RP-Advs" (Mensagens de Advertência) para o BSR do domínio, contendo seus endereços e o endereço do grupo que pretende ser o RP. O BSR então define qual será o RP e divulga dentro do domínio. Desta forma os DRs podem identificar onde está o RP.

Os *hosts* que desejam fazer parte de um grupo mandam mensagens explícitas de associação aos roteadores que estão no caminho do RP escolhido. Para isso, ele cria um cartão com informações para a entrada *multicast*, denominada aqui como entrada (*,G), por onde serão enviados os pacotes.

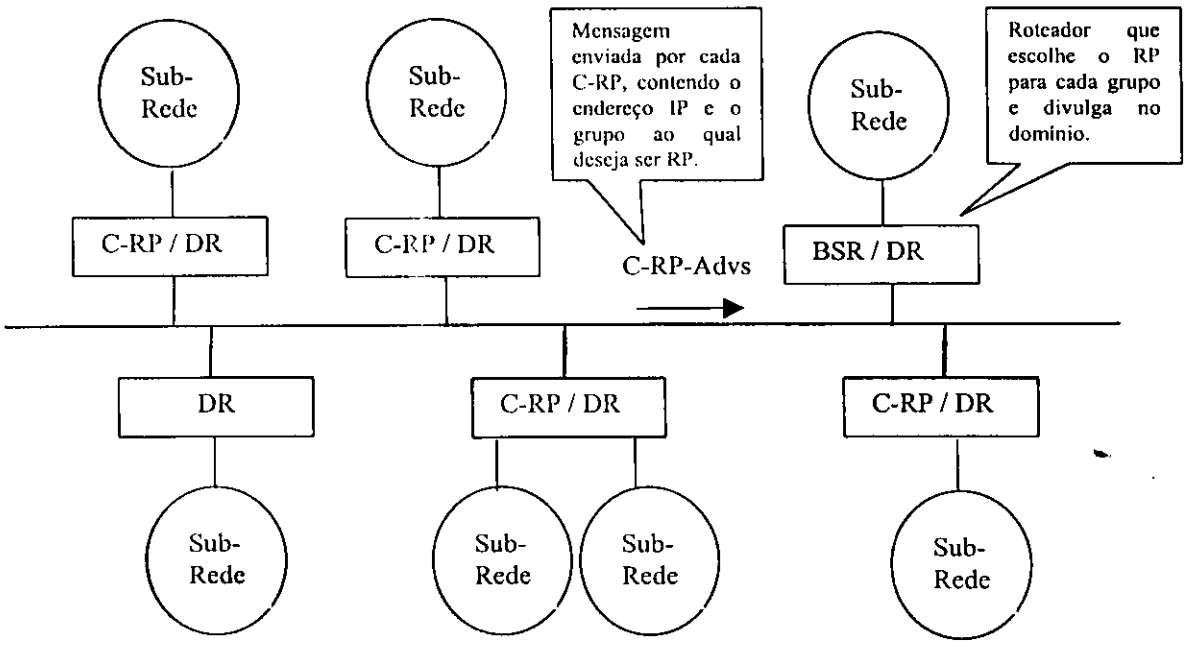


Figura 28. Funcionamento do mecanismo *Bootstrap*

A entrada (*,G) contém o endereço do RP e do grupo *multicast*, a interface de saída que é definida pela mensagem "*Host-Membership-Report*" recebida do novo membro, a interface de chegada que é definida pela interface usada para enviar pacotes *unicast* ao RP, o

bit de controle RPT (RP-*Tree*) ajustado para 1, indicando que será usado a árvore RP e o bit de controle WC (*WildCard*) ajustado para 1, indicando que os receptores abaixo do DR esperam todos os pacotes, de todas as fontes, via árvore RP. O cartão é mostrado nas ações 3, 5 e 7 da Figura 29.

Cada roteador até o RP cria ou atualiza sua entrada (*,G) quando recebe uma mensagem de "*Junção*" com os bits RPT e WC inicializados em 1. A interface pela qual a mensagem de "*Junção*" chega é adicionada à lista de interfaces de saída; baseado nisso, cada roteador, entre o receptor e o RP, envia uma mensagem de "*Junção*", especificando o RP. O conteúdo do pacote contém o Endereço *Multicast* = G, Junção {RP,WC,RPT} e Poda = Nula, como mostrado nas ações 4 e 6 na Figura 29.

Quando não existirem mais membros locais conectados ao grupo, o DR fica sabendo através do IGMP. Se não houver *downstream* necessitando de pacotes do par (Grupo, Fonte), a entrada (*,G) é removida.

Quando um *host* inicia a transmissão de pacotes de dados *multicast* para um grupo, inicialmente, seu DR entrega cada pacote ao RP, encapsulando-os em mensagens "Registradas" e os envia para o RP daquele grupo na forma *unicast*. O RP desencapsula as mensagens "Registradas" e envia os pacotes de dados para os membros do grupo, pela árvore RP.

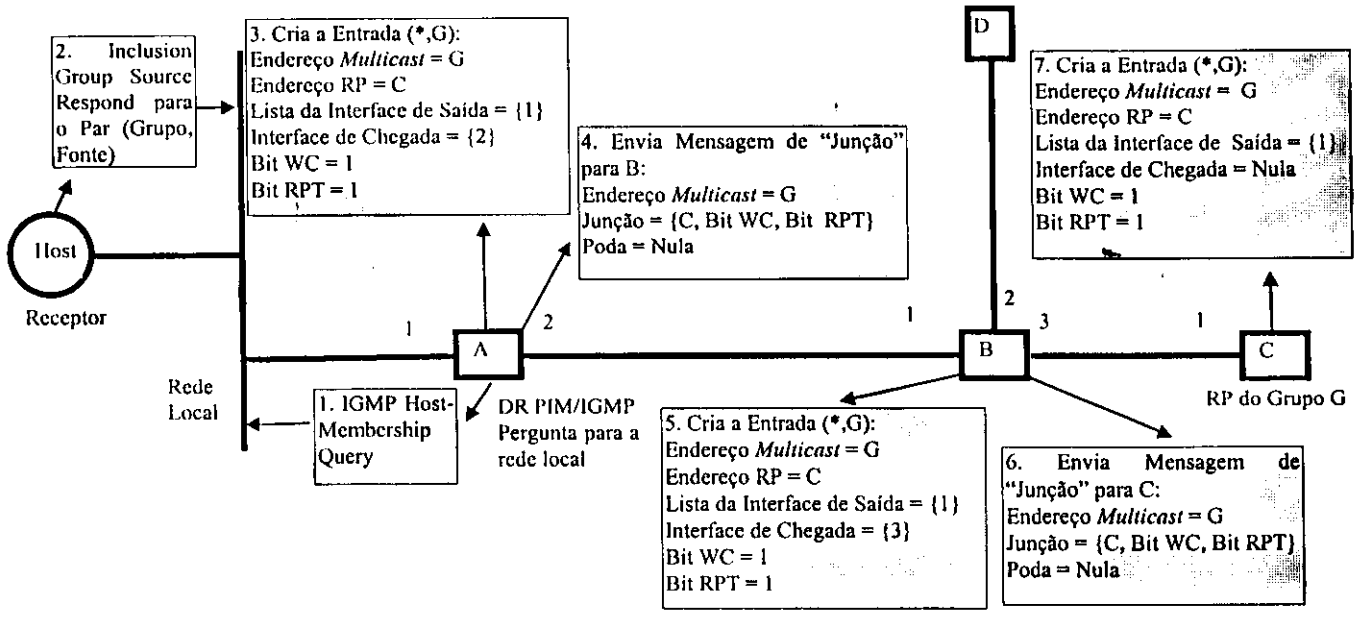


Figura 29. Exemplo de como um receptor junta-se e ajusta-se à árvore RP

Para aumentar a eficiência do protocolo, quando uma fonte está transmitindo um número significativo de pacotes de dados, os roteadores com receptores locais podem juntar-se a esta fonte específica, pela árvore SPT, criando a entrada (S,G) e enviando mensagens de "Junção" pela mesma.

O DR da fonte pode parar de encapsular os pacotes de dados em mensagens "Registradas", quando ele receber do RP, uma mensagem "*Register-Stop*". A política recomenda, inicialmente, que seja feita a troca para a árvore SPT, após receber um número significativo de pacotes de dados, de uma fonte particular, durante um intervalo específico de tempo. Para realizar essa política, o roteador monitora os pacotes de dados gerados pela fonte, obtendo a taxa de dados produzida, oportunizando a tomada da decisão sobre qual a árvore de distribuição que será usada, como mostra a Figura 30, onde o roteador C inicializa um estado (S,G).

Quando uma entrada (S,G) for ativada, uma mensagem de "Junção" será enviada para a fonte S, sendo que S está na lista de Junção[6], como mostra as ações 2 e 4 da Figura 30. Quando a entrada (S,G) é criada, a interface de saída é copiada de (*,G), isto é, todos os galhos de compartilhamento da árvore local são repetidos na árvore SPT. Nesse sentido, quando um pacote de S chega e condiz com essa entrada, todos os receptores continuarão a receber os pacotes da fonte ao longo desse caminho.

Note que o estado (S,G) será mantido em cada roteador que se liga diretamente a um receptor, sendo responsável por iniciar e manter uma árvore SPT. Mesmo quando (*,G) e (S,G) se sobrepõem, ambos os estados são necessários para produzir as mensagens de "Junção/Poda" a uma específica fonte. Um temporizador "*Entry-timer*" é ajustado para a entrada (S,G), que será apagada quando o tempo "*Entry-timer*" expirar.

[6] Lista de Junção é uma das duas listas de endereço IP *unicast* que está incluida em uma mensagem de Junção/Poda. Cada endereço refere-se a uma fonte ou RP. Ela indica as fontes ou RPs que um ou mais receptores desejam juntar-se.

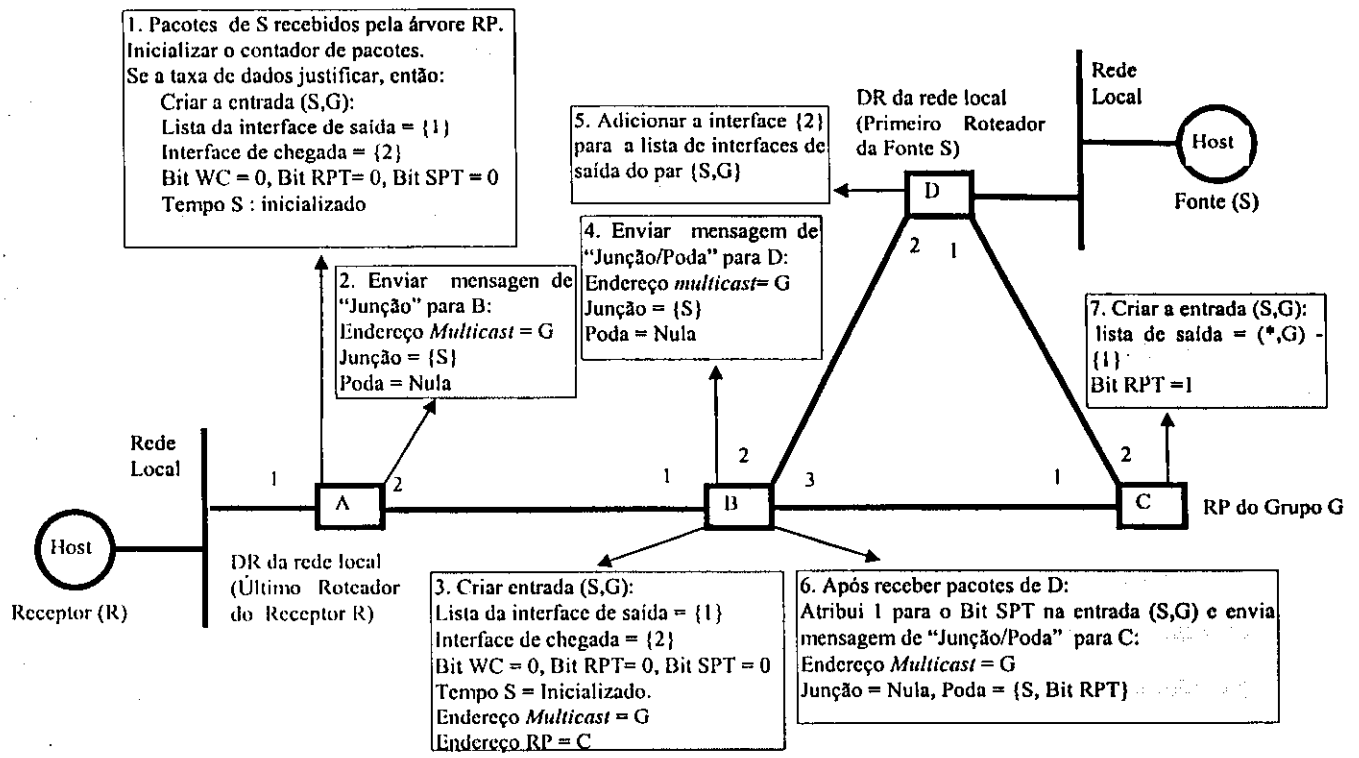


Figura 30. Exemplo da troca para a árvore SPT

Somente o RP pode iniciar a troca para a árvore SPT. Conseqüentemente, roteadores com membros locais criam o estado (S,G) em resposta a pacotes de dados da fonte (ação 1 da Figura 30), enquanto que roteadores intermediários somente criam o estado (S,G) em resposta a mensagens de "Junção" de outros roteadores a baixo e que tenham S na lista de saída (ação 3 da Figura 30).

A entrada (S,G) é inicializada com o "bit SPT" vazio, indicando que os galhos da árvore SPT para S, ainda não foram ajustados completamente e o roteador pode aceitar ainda pacotes de S que cheguem pela entrada (*,G).

Quando um roteador, com uma entrada (S,G) e um "bit SPT" vazio, inicia a recepção de pacotes de uma nova fonte S, pela entrada (S,G), e esta por sua vez difere da interface de chegada da entrada (*,G), o roteador ajusta o bit SPT e envia uma mensagem de "poda" para o RP, indicando que não quer mais receber pacotes S pela árvore RP. A mensagem de "Poda" enviada para o RP inclui S na lista de Poda, e o "*bit* RPT" inicializado, indica que os pacotes de S não deverão ser enviados por esse ramo da árvore (ação 6 da Figura 30).

Para que um novo membro possa associar-se a um determinado grupo, no qual o estado de "poda" tenha sido estabelecido, é necessário que o estado atual seja erradicado, desta forma, quando uma junção (*,G) chegar a um roteador que tenha alguma entrada (Si,G)RPT-bit (isto é, entrada que causa transmissão de "poda" para o RP), esta deve ser atualizada no roteador, possibilitando que os pacotes cheguem até novos membros.

O PIM-SM pode interoperar com outros protocolos *multicast*, através dos PMBRs (PIM *Multicast Border Routers*), os quais sabem que ao receberem pacotes de dados endereçado a uma entrada (*,*,RP), devem procurar o RP para esse pacote em um outro domínio. Todos os roteadores PIM podem ser capazes de suportar o estado (*,*,RP) e interpretar mensagens de "Junção/Poda" [HAWK 97]. Para maiores informações sobre o protocolo PIM-SM consulte [RFC 2117].

3.6. *CORE BASED TREE*

O protocolo CBT foi concebido com a intenção principal de aumentar a escalabilidade apresentada nos protocolos que constróem árvores de roteamento para cada fonte *multicast*. A abordagem consiste em implementar uma única árvore de roteamento para cada grupo, que será utilizada por todas as fontes do grupo [RFC 2201].

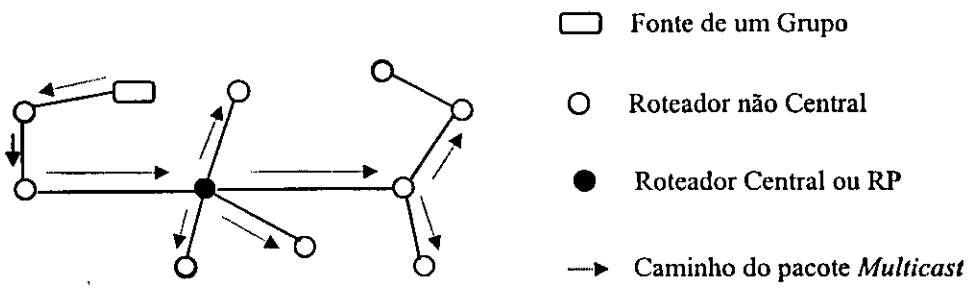


Figura 31. Árvore de Entrega *Multicast* CBT

As fontes de um determinado grupo enviam seus dados para o roteador central (também chamado de *Rendezvous Point*), como se estivessem enviando pacotes *unicast*. Quando estes pacotes chegam ao roteador central, são disseminados aos demais roteadores pertencentes ao grupo, que por sua vez usam o identificador de grupo (número IP do grupo), como um índice para encontrarem dentro do *Forwarding Cache* as interfaces a quem devem entregar o pacote.

A escolha do roteador central pode ser definida de duas formas:

• Pela configuração manual dos roteadores folhas, para reconhecerem o roteador central através da especificação de mapeamento (roteador central, grupo); ou

• Através de um mecanismo de eleição chamado "*Bootstrap*", que define a configuração de alguns roteadores como "roteadores centrais candidatos" dentro de um domínio, onde são eleitos dinamicamente, como foi especificado no PIM-SM. A diferença é que o PIM-SM mantém um único RP por domínio, enquanto que no CBT os RPs são definidos por grupos.

O CBT constrói e mantém uma árvore de distribuição *multicast* somente para alcançar redes que possuem membros associados a um grupo. Para alcançar este objetivo, os *hosts* expressam seu interesse em juntar-se a um grupo, através de respostas a perguntas IGMP feitas pelo DR (*Designated Router*).

O DR é escolhido através de mensagens "*Hello*", com TTL igual a 1, e especificam a sua prioridade que vai de 1 até 245. A prioridade de valor menor é que define o DR, sendo que, em caso de empate, o roteador com menor endereço IP prevalece. As mensagens "*Hello*" são geradas a cada 60 segundos (*Default*).

O roteador ao receber as respostas IGMP, gera uma mensagem de "Requisição de Junção" (*Join_Request*), que é enviada até o próximo roteador no caminho do roteador central. Essa mensagem de junção pode ser explicitamente reconhecida pelo roteador central ou por outro roteador que esteja no caminho, entre o roteador de origem da "Requisição de Junção" e o roteador central.

Para a confirmação da "Requisição de Junção" é usada uma mensagem de "Confirmação de Junção" (*Join_Ack*), que é enviada de volta ao roteador que gerou a "Requisição de Junção". Caso a mensagem de "Confirmação de Junção" não chegue dentro de um tempo pré-definido (*Rtx_Interval* = 5 segundos)[7], ocorrerá retransmissões da

[7] Todos os valores especificados juntamente com as variáveis definidas são apresentados como *default* na [RFC 2189].

"Requisição de Junção" até que ocorra a junção, ou o tempo (*Transient_Timeout* = 1.5 * *Rtx_Interval*) se esgote.

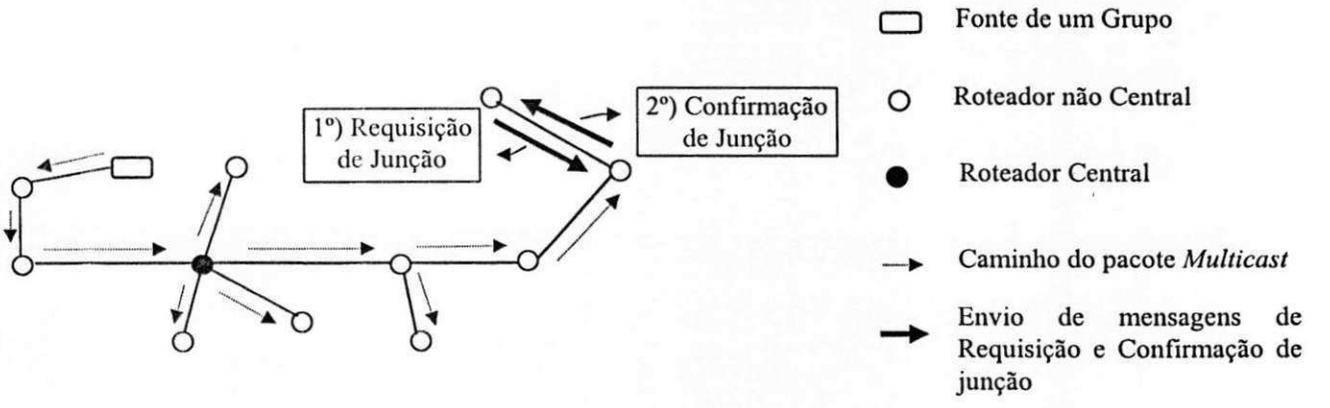


Figura 32. Roteador requerendo associação à árvore CBT

Estas etapas servem para definir a árvore de roteamento *multicast* relacionada ao grupo especificado e ajustar o *Forwarding Cache*, definindo o grupo, a interface de chegada e as interfaces de saída, conhecidas no CBT como pai e filhos respectivamente.

Todos os roteadores pertencentes a uma árvore CBT, exceto o roteador central, devem manter seu pai informado, à respeito da necessidade de receber pacotes *multicast* de um determinado grupo. Esta informação é obtida pelo envio periódico de mensagens do tipo (*Echo_Request*), vindas dos roteadores filhos ao roteador pai, a cada (*Echo_Interval* = 60 segundos). Caso não sejam recebidas mensagens durante este intervalo de tempo, inicia-se o processo de "poda".

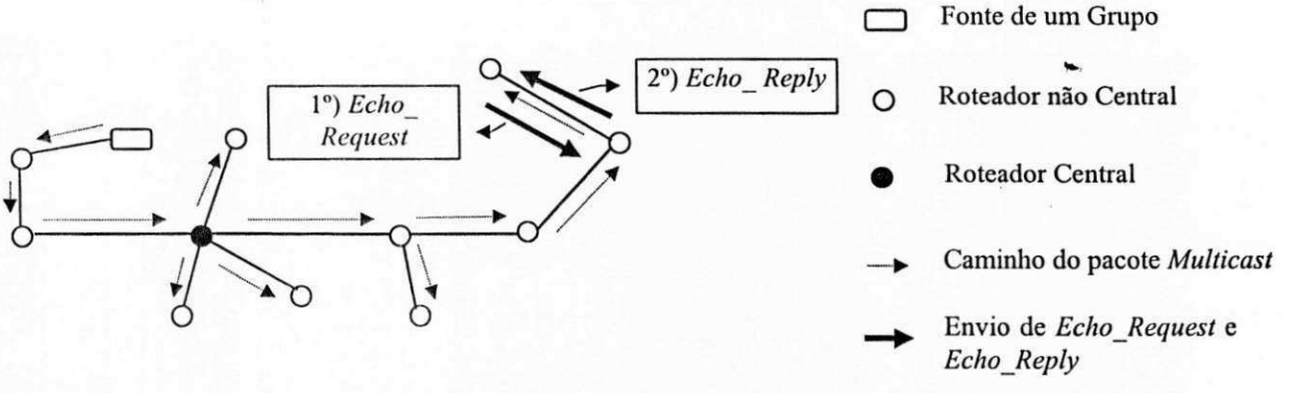


Figura 33. Roteador informando permanência na árvore CBT

O roteador pai, ao receber a mensagem (*Echo_Request*) gera uma resposta (*Echo_Reply*), que contém a lista de grupos que a interface filho pertence. Caso o (*Echo_Reply*) não seja recebido em (*Group_Expire_Time* = 1,5 * *Echo_Interval*), são geradas mensagens de (*Quit_Notification*), descritas a seguir.

As notificações de "poda" são definidas pelos roteadores filhos, quando enviam uma mensagem de "notificação de saída" (*Quit_Notification*), indicando que não possuem interesse em receber pacotes de um determinado grupo. Estas mensagens não possuem confirmação, desta forma, para manter a consistência, são enviadas 3 vezes, durante intervalos de tempo (*HoldTime* = 3 segundos).

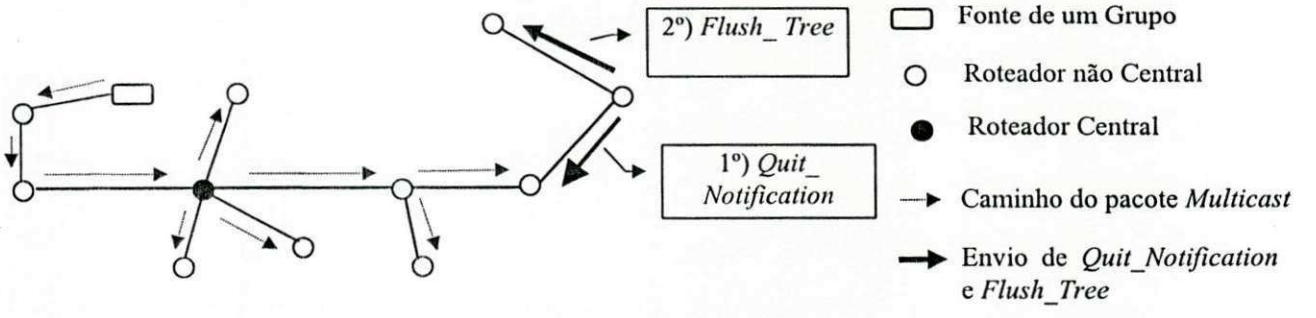


Figura 34. Roteador informando saída da árvore CBT

A "notificação de saída" gera mensagens (*Flush_Tree*), que são enviadas do roteador que notificou a saída do grupo a todos os seus filhos (*downstream*), indicando a eles que não receberão mais informações a respeito de um determinado grupo.

Maiores informações a respeito do protocolo CBT, são encontras em [RFC 2189] e [RFC 2201].

3.7. COMPARAÇÃO DOS PROTOCOLOS DE ROTEAMENTO *MULTICAST*

Nesta seção apresentamos comparações dos protocolos *multicast* de acordo com as técnicas de construção de árvores *multicast* e controle de tráfego, além de abordarmos as vantagens e desvantagens de cada protocolo descrito nesse capítulo, através das informações obtidas a partir de trabalhos realizados por outros autores.

Primeiramente devemos distinguir as três técnicas usadas para a implementação das árvores de roteamento *multicast*. O DVMRP, MOSPF e PIM-DM sempre utilizam árvores SPT para a entrega dos pacotes, desta forma a fonte de um grupo sempre é a raiz da árvore. Por outro lado, o CBT utiliza a árvore RP, na qual a raiz da árvore é o RP, que centraliza todo o tráfego *multicast*. Já o PIM-SM utiliza um método híbrido, utilizando primeiramente uma árvore RP, e após uma certa quantidade de tráfego gerado pela fonte monta a árvore SPT.

Conforme avaliações de [CALV 94] "os protocolos que utilizam árvores SPT apresentam menor atraso fim-a-fim em relação aos protocolos que montam suas árvores de entrega *multicast* a partir de um RP, devido à utilização do menor caminho entre a fonte e o destino *multicast*".

Conforme [FARR 97] "o protocolo CBT, baseado na árvore RP, apresenta maior concentração de tráfego *multicast* nos enlaces que ligam o RP aos demais roteadores, devido à utilização constante dos mesmo para a entrega dos pacotes *multicast*. Em contrapartida, o PIM-DM (utiliza árvore SPT) e o PIM-SM (híbrido) apresentam um nivelamento maior em relação à ocupação dos enlaces, devido à construção de árvores distintas para cada fonte".

De acordo com [RFC 2201], um dos estímulos para à construção do protocolo CBT é que "as árvores SPT apresentam problemas de escalabilidade, devido a construção de uma árvore *multicast* para cada fonte de um grupo". O impacto disso reflete-se na manutenção e processamento das tabelas de rotas, onde existem muitos grupos e fontes simultâneos. A Figura 35 demostra o exposto acima.

Número de Grupos	10			100			1.000		
Tamanho dos Grupos	20			40			60		
Percentual de Fontes ativas por Grupo	10%	50%	100%	10%	50%	100%	10%	50%	100%
Número de Entradas na Tabela da árvore SPT	20	100	200	400	2.000	4.000	6.000	30.000	60.000
Número de entradas na Tabela da árvore RP	10			100			1.000		

Figura 35. Comparação de entradas nas tabelas de rotas das árvores SPT e RP.

Outro fato que deve ser considerado está relacionado com o controle de tráfego, que pode ser por grupo ou por fonte. No caso de ser por grupo todos os destinos recebem os

pacotes gerados para o grupo, como é o caso do CBT. Entretanto, se o tráfego for diferenciado por fonte, um determinado destino pode escolher de quais fontes quer receber pacotes, como é o caso do DVMRP, MOSPF, PIM-DM e PIM-SM.

A motivação da construção do CBT somente pelo controle de grupo, respalda-se no fato de que a maioria dos destinos requerem o tráfego de todas as origens, e a manutenção da escalabilidade imposta pela árvore RP [RFC 2201].

Segundo [FARR 97], "os protocolos baseados na árvore SPT geram maior número de pacotes de sinalização que o CBT, devido às inúmeras mensagens de "podas" em galhos inúteis, decorrentes do maior número de árvores construídas";

Outras consideração levantada por [FARR 97] indicando aumento na geração de pacotes de sinalização, está relacionada aos protocolos de modo denso, como é o caso do DVMRP, MOSPF e PIM-DM. "Estes inundam todas as interfaces ao estabelecer uma nova fonte *multicast*, ficando propensos a receber maior número de mensagens de "poda" que os protocolos de modo esparso, os quais recebem mensagens explicítas dos destinatários indicando que desejam juntar-se a um determinado grupo, como é o caso do CBT e PIM-SM";

Alem das considerações acima, abordaremos vantagens e desvantagens particulares de cada protocolo.

☛ DVMRP – Vantagens

• "Aceita a implementação de túneis para interligar domínios que suportam *multicast*, passando através de domínios que não suportam *multicast*" [RFC 1075];

• "Muito utilizado por ser o protocolo implementado no *Backbone* do MBONE" [ERIK 97];

☛ DVMRP – Desvantagens

• Depende de informações obtidas do RIP, inviabilizando sua implementação em conjunto com outros protocolos de roteamento *unicast* [RFC 1075];

• Atualmente, não distingue hierarquias de domínios. De acordo com [THYA 96] "se o MBONE continuar crescendo exponencialmente, em pouco tempo os roteadores executando o DVMRP sofrerão um colapso, devido à falta de memória ou recursos de processamento, para manter uma entrada em suas tabelas para cada rede que compõem o MBONE";

• Por se tratar de um protocolo *Vector-Distance* apresenta os mesmos problemas de convergência do RIP. No momento em que um enlace sofre algum tipo de problema que inviabiliza a sua utilização, os roteadores levam um tempo elevado para detectar o problema e utilizar rotas alternativas, com isso propiciando a perda de pacotes [PUSAT 96].

☛ **MOSPF – Vantagens**

• "Define o escopo de um sistema autônomo, tratando somente as informações que são relevantes a este sistema, o que o torna um protocolo modular" [RFC 1584]. Esta definição do MOSPF é importante no que diz respeito a manutenção e processamento das tabelas de rotas, já que estas precisam ser atualizadas somente com informações internas ao sistema autônomo. As informações externas ao sistema autônomo são tratadas por roteadores específicos, somente quando forem necessárias;

• "Pode suportar *multicast* em um mesmo domínio, mesmo que nem todos os roteadores estejam implementados com MOSPF" [SEMER 97]. Esta característica possibilita a implementação *multicast* de forma gradativa.

☛ **MOSPF – Desvantagens**

• Depende de informações obtidas do OSPF, inviabilizando sua implementação em conjunto com outros protocolos de roteamento *unicast* [RFC 1584];

• "O MOSPF não possui habilidades para implementação de túneis, que possam enviar pacotes *multicast* através de roteadores que não estão configurados para suportar *multicast*" [RFC 1585]. Esta limitação inviabiliza a implantação do MOSPF como protocolo padrão no *backbone* do MBONE, já que os *sites* que fazem parte do MBONE atualmente estão cercados por redes que não estão aptas a processar pacotes *multicast*.

☛ **PIM-DM – Vantagens**

• "Independe de protocolos de roteamento *unicast*" [HELMY 97];

• "O PIM-DM assume que todas as interfaces *Downstream* gostariam de receber os pacotes gerados por uma fonte, o que é considerado bom para grupos densamente populosos" [HELMY 97], já que se espera que todas as sub-redes em sistemas autônomos tenham membros de um determinado par (Grupo, Fonte).

☛ **PIM-DM – Desvantagens**

• "Sub-redes que possuem mais de uma interface de entrada, recebem pacotes duplicados ao serem estabelecidas as árvores *multicast*, devido à inundação gerada na rede para a descoberta das rotas até cada destinatário de um par (Grupo, Fonte)" [HELM 97].

• "Gera mais pacotes de sinalização do que qualquer outro protocolo *multicast*, decorrente de inúmeras "podas" que precisam ser geradas ao serem construída as árvores *multicast*" [FARR 97].

☛ **PIM-SM – Vantagens**

• "Cria as árvores *multicast* a partir de solicitações explícitas dos roteadores que desejam receber pacotes de um determinado par (Grupo, Fonte). O que é considerado bom para grupos esparsamente distribuídos, já que nem todas as sub-redes em um sistema autônomo possuem membros de um determinado par (Grupo, Fonte)" [RFC 2117].

• "Por se tratar de um protocolo que trabalha com uma técnica híbrida utilizando árvores RP e árvores SPT, pode viabilizar um atraso fim-a-fim aceitável, diminuindo os problemas de escalabilidade" [FARR 97].

☛ **PIM-SM – Desvantagens**

• "Dificuldade de implementação em decorrência da troca da árvore RP para a árvore SPT" [FARR 97].

• "Não permite mais de um RP em um sistema autônomo" [FARR 97].

☛ **CBT – Vantagens**

- Não apresenta problemas de escalabilidade, como mostramos na Figura 35;

- "Definir RPs diferentes para grupos diferentes, oportunizando um melhor balanceamento do tráfego nos enlaces " [FARR 97].

☛ **CBT – Desvantagens**

- "Uma das desvantagem do CBT é o atraso fim-a-fim, decorrente da utilização da árvore RP, que nem sempre utiliza o melhor caminho para a entrega dos pacotes *multicast*"

- "Potenciais congestionamentos em algumas aplicações devido à concentração de tráfego para muitas fontes no mesmo enlace, e vulnerabilidade para falhas no RP" [CALV94].

CAPÍTULO 4

O SIMULADOR DE PROTOCOLOS *MULTICAST* (SPM)

A avaliação de desempenho de um sistema pode ser realizada através de medições feitas no sistema em produção, ou através de um modelo que represente o sistema englobando suas características e funcionalidades (modelagem matemática).

Considerações feitas após o sistema estar em produção nos permite obtenção de valores exatos, entretanto, em muitos casos, é importante que a avaliação do sistema seja feita a nível de projeto, viabilizando a redução de custos e contribuindo para a correção de erros.

Para solucionar modelos matemáticos podem ser utilizadas as técnicas analíticas, ou as técnicas numéricas, ou ainda uma solução híbrida que envolva as duas técnicas mencionadas [MOURA 86].

Uma solução analítica é obtida através de resultados de equações matemáticas que

relacionam os parâmetros do modelo com as medidas de desempenho de interesse, como por exemplo, algoritmos baseados na Teoria das Filas. Essa solução nem sempre pode ser empregada devido à complexidade do modelo e, quando empregada, na maioria das vezes, exige suposições simplificadas para o sistema a ser modelado.

A solução numérica é obtida através de investigações, métodos de convergência, interpolação, sendo fornecido um valor estimado, simultaneamente ao erro do método, como por exemplo a Simulação Digital. Essa solução pode ser aplicada a qualquer modelo, sem que haja restrições quanto a sua complexidade, entretanto os modelos devem ser construídos incorporando somente as características mais importantes de um sistema, para facilitar a sua construção [MOURA 86].

A Simulação Digital é a técnica numérica mais utilizada para realizar experimentos em um modelo a fim de obter as medidas de desempenho de interesse. Para facilitar o uso da simulação digital, foram construídas ferramentas para a simulação de modelos, como por exemplo: BONeS DESIGNER (Block Oriented Network Simulator) [ALTA 95], SAVAD (Sistema de Avaliação de Desempenho de Modelos de Redes de Filas) [SOUTO 94], entre outros.

Neste contexto, enfocamos neste trabalho o SPM, que foi desenvolvido com o objetivo de executar simulações de modelos, embasados nos protocolos de roteamento *multicast* como o MOSPF, DVMRP, PIM-SM, PIM-DM e CBT.

Neste capítulo, descrevemos a implementação do SPM, bem como sua utilização para simular modelos de redes de computadores com protocolos de roteamento *multicast*.

4.1. A ESTRUTURA DO SPM

As redes para o qual o SPM foi projetado apresentam os *hosts* interligados localmente através dos padrões *Ethernet*, *Token Ring* e ATM (*Asynchronous Transfer Mode*), tendo como ponto de partida para interligação com outras redes, roteadores que passam os dados para seus vizinhos, de acordo com a determinação de um protocolo de roteamento *multicast*, como mostra a Figura 36.

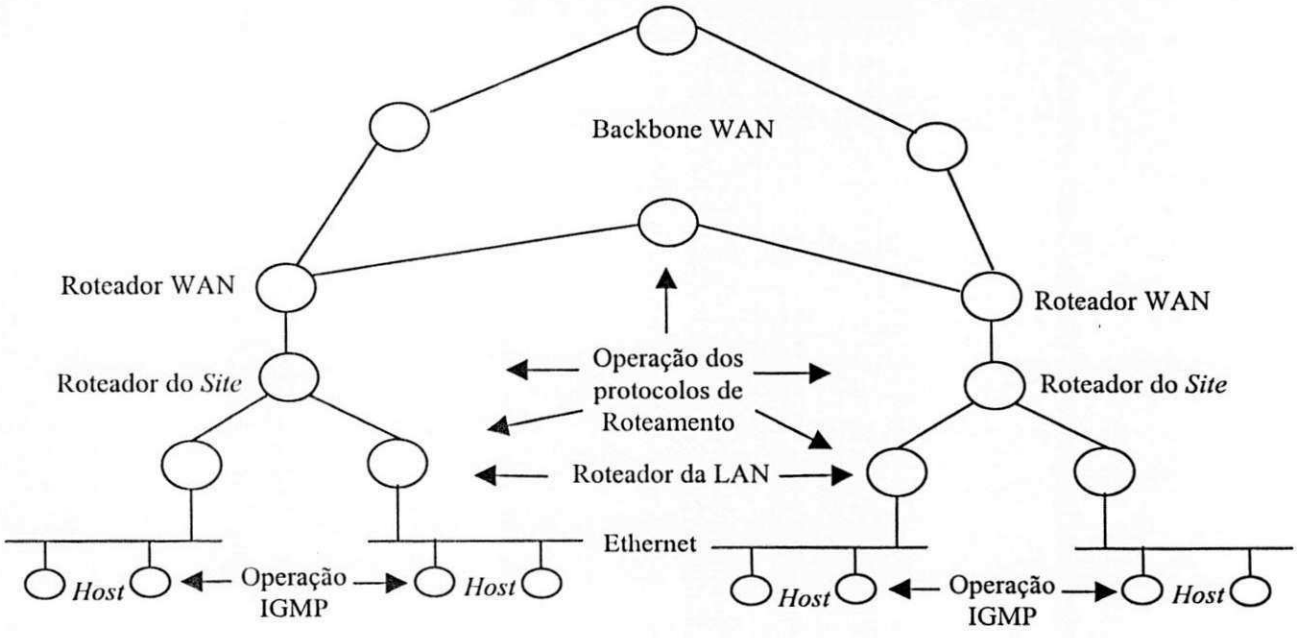


Figura 36.Estrutura das redes IP

Para a construção do SPM foram abordadas três representações distintas de acordo com a Figura 36. A primeira é o funcionamento das redes locais e a operações do IGMP, a segunda é o funcionamento dos roteadores e a atuação dos protocolos *multicast* sobre eles e, a terceira, é o fluxo de pacotes na rede.

Para que o simulador pudesse absorver essas representações, este foi projetado de forma modular, onde os módulos assumem papéis específicos e interagem entre si durante a execução da simulação.

A estrutura do simulador está organizada em nove módulos lógicos:

- **Entrada de Dados**: é responsável por absorver informações externas ao simulador, para definição dos parâmetros envolvidos na simulação;

- **Controle de Simulação**: é responsável pelo encaminhamento dos pacotes das origens para os respectivos destinos;

- **Controle de Tempo**: controla a geração de números aleatórios, para a definição dos pulsos do relógio do simulador;

• **Controle de Pacotes**: controla a geração dos pacotes em cada nodo da rede. Nesse modulo são definidos todos os campos internos do pacote;

• **Ambiente de Rede**: define a topologia da rede;

• **Biblioteca de Modelos**: fornece as regras para a entrega dos pacotes de acordo com a especificação do protocolo *multicast* envolvido na simulação. Os modelos do DVMRP, MOSPF, PIM-DM, PIM-SM e CBT foram programados em linguagem "C" e anexados às bibliotecas do simulador para execução das simulações;

• **IGMP**: contém as regras de gerência de grupos, definidas pelo protocolo IGMP;

• **Pontos de Investigação**: são responsáveis pela coleta das informações que devem ser avaliadas;

• **Saída de Dados**: é o responsável em mostrar os resultados armazenados nos pontos de investigações;

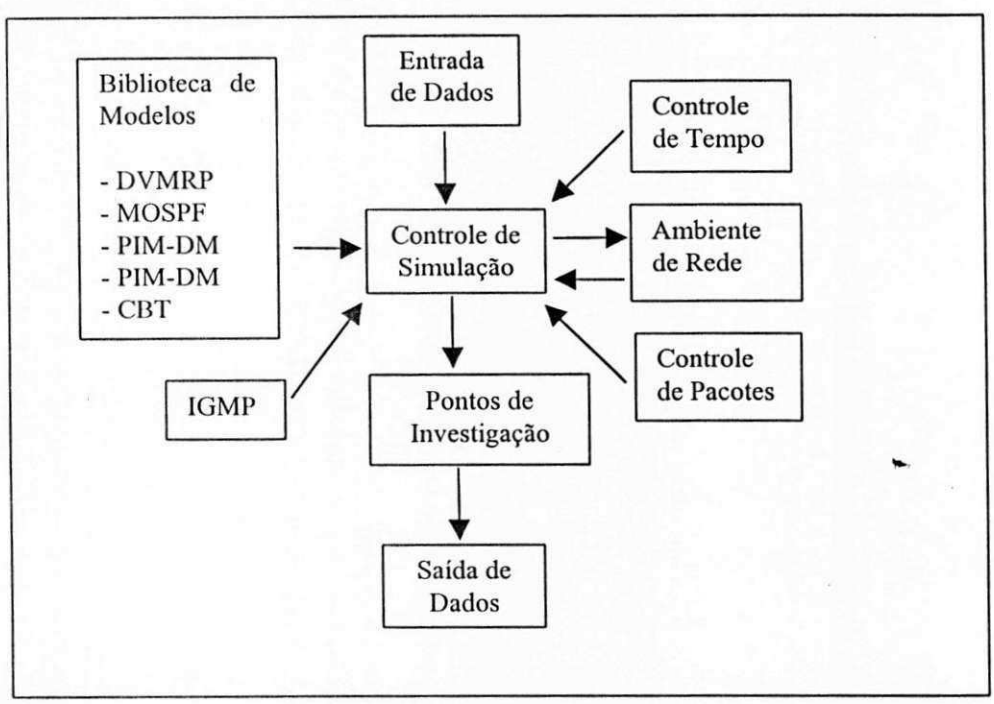


Figura 37. Fluxo de operação entre os módulos do SPM

4.2. PRINCIPAIS ESTRUTURAS DE DADOS DO SPM

De acordo com [TANEN 95], estruturas de dados retratam as relações existentes entre os dados, de modo análogo ao uso de um modelo matemático para espelhar alguns aspectos de uma realidade física. Desta forma, nessa seção apresentamos as estruturas de dados utilizadas no SPM, para demostrar como definimos o problema real em termos computacionais.

4.2.1. ESTRUTURA PACOTE

Esta estrutura representa os pacotes de dados e sinalização, definindo as informações usadas pelo simulador para controlar cada pacote.

```
typedef struct pacote {
   int       fonte;       // define a fonte que criou o pacote
   int       destino;     // define o destino do pacote (outro roteador ou grupo)
   int       tamanho;     // define o tamanho do pacotes em octetos
   long int  tempo_cria;  // armazena o instante da criação do pacote
   long int  atraso;      // armazena o atraso sofrido pelo pacote para alcançar seu destino
   char      tipo;        // define se o pacote é de dados ou sinalização
} T_pacote;
```

4.2.2. ESTRUTURA FORWARDING_CACHE

Esta estrutura é uma lista encadeada definida, para cada roteador, permitindo que este saiba todos os pares (Grupo, Fonte) existentes. Além disso, permite a identificação dos pares (Grupo, Fonte) a que o roteador está associado no momento.

```
typedef struct cache {
   int              grupo;        // identifica o grupo
   int              fonte;        // identifica a fonte
   char             adesao;       // identifica a associação ao par (Grupo, Fonte)
   struct cache     *P_proximo;   // liga uma estrutura a outra
   struct enlace    *cab_enlace ; // aponta para o enlace a ser utilizado para enviar os pacotes
```

```
} T_cache;
```

4.2.3. ESTRUTURA ÁRVORE

Esta estrutura é utilizada para definir a árvore de roteamento *unicast* e *multicast*. É através dela que os pacotes se deslocam de um ponto para outro na rede.

```
typedef struct arvore {
   int              roteador;      // identifica o roteador
   struct arvore *pai;             // identifica o roteador upstream
   struct arvore *filho;           // identifica o roteador downstream
   struct arvore *irmao;           // usado para achar os outros roteadores downstream
   int              nivel;         // identifica a posição do roteador na árvore
   T_cache   *P_cache;             // apontador para o Forwarding Cache associada ao roteador
} T_arvore;
```

4.2.4. ESTRUTURA LISTA DE FONTES

Esta estrutura define todas as fontes que existem em um determinado instante da simulação. Ela aponta para a árvore de entrega *multicast* que deve ser utilizada pela fonte para enviar seus pacotes.

```
typedef struct lista_f{
   int              fonte;              // identifica a fonte
   long int         tempo_criacao; // armazena o instante de criação da fonte
   struct lista_f *p_proximo;     //  apontador para a próxima fonte da lista
   T_arvore      *cabArvoreM;  //  apontador para a árvore usada para entrega dos pacotes
} T_fonte;
```

4.2.5. ESTRUTURA GRUPOS

Esta estrutura representa todos os grupos existentes e aponta para a lista de fontes pertencentes a um grupo.

```
typedef struct {
   int          grupo;    // identifica o grupo
```

```
    T_fonte    *fontes; // apontador para a lista de fontes do grupo
} T_grupo;
```

4.2.6. Estrutura Ocupação dos Enlaces

Esta estrutura identifica os pacote que estão utilizando um enlace, definindo o tipo de pacote e quanto tempo ele necessita para desocupar o enlace.

```
typedef struct ocupacao {
    long int          tempo_exclui; // armazena o instante de remoção da estrutura
    char              tipo;         // define o tipo de pacote (dados ou sinalização)
    struct ocupacao *p_proximo;     // apontador para a próxima estrutura
} T_ocupacao;
```

4.2.7. Estrutura Enlaces

Esta estrutura identifica os enlaces que foram definidos e a quantidade de pacotes de dados e de sinalização que estão trafegando dentro deles em um determinado instante. Juntamente com o controle de *buffer*, nos permite o cálculo do descarte de pacotes (*overflow*).

```
typedef struct enlace{
    int          numero;        // identifica o enlace
    int          roteadorA;     // identifica um dos roteadores interligados
    int          roteadorB;     // identifica o outro roteador
    long int     pacdados;      // identifica o número de pacotes de dados no enlace
    long int     pacsinalizacao; // identifica o número de pacotes de sinalização no enlace
    struct ocupacao *cabpacote; // apontador para  a lista de ocupação dos enlaces
} T_enlace;
```

4.2.8. Interação Entre as Estruturas de Dados

As interações entre as estruturas de dados determinam a execução do sistema e a obtenção dos resultados. A definição dos caminhos entre os membros de um Par (Grupo, Fonte) podem ser definidos através da interação entre a estrutura de Grupos com a estrutura

de Fontes, que por sua vez liga-se a uma estrutura Árvore que será usada para montar os caminhos de entrega *multicast*.

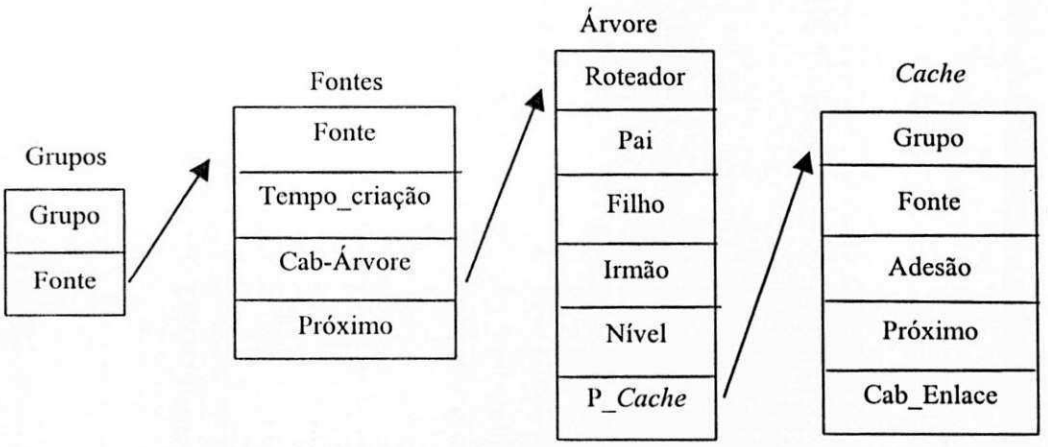


Figura 38. Definição dos caminhos entre os membros de um par (Grupo, Fonte)

Os pacotes após serem gerados devem ser colocados nos enlaces, para percorrerem seus caminhos até seus destinos, guiados pela estrutura *Cache*. Na Figura 39 apresentamos a interação entre as estruturas que fazem este serviço.

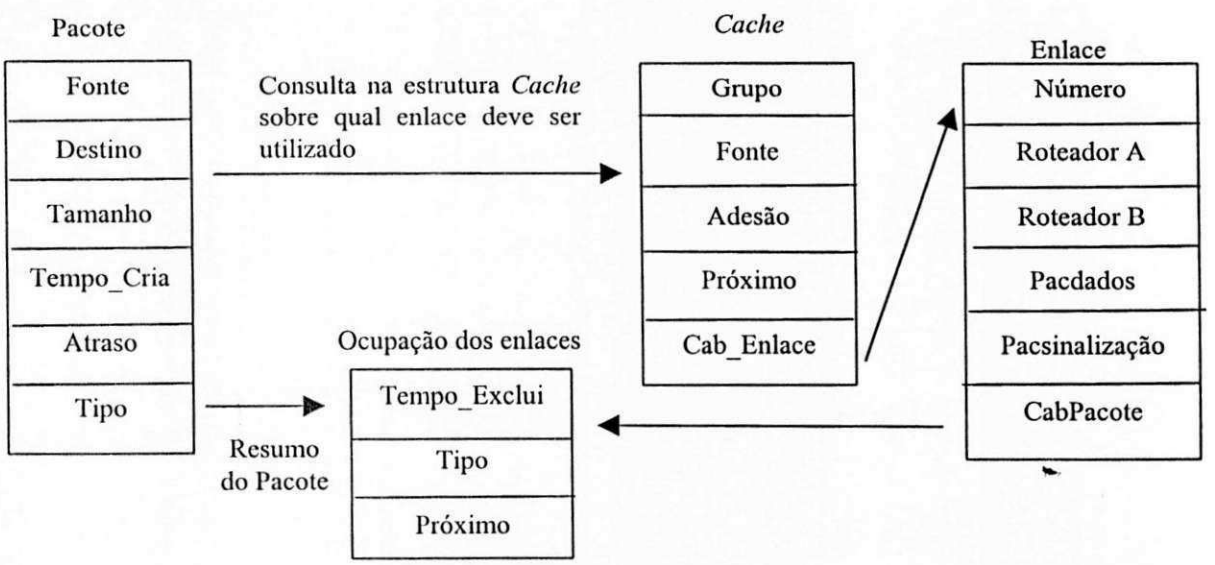


Figura 39. Transferência dos pacotes através dos enlaces

4.3. AMBIENTE DE DESENVOLVIMENTO DO SPM

Para o desenvolvimento do SPM, escolhemos a linguagem de programação C (compilador *Boorland* C), devido aos recursos de manipulação direta de *bits*, *bytes*,

palavras e ponteiro, o que a torna capacitada para a programação de sistemas, onde essas operações são comuns [SCHIL 91].

Outro ponto decisivo na escolha da linguagem de programação C foi o fato de que "seu principal componente estrutural é a função, a qual admite que sejam definidas e codificadas separadamente as diferentes tarefas de um programa, permitindo que a programação seja definida de forma modular" [SCHIL 91].

Os equipamentos utilizados para o desenvolvimento do SPM foram micro computadores compatíveis com a linha IBM PC, executando o sistema operacional Windows 95, por serem os mais usados em todo o mundo, possibilitando que o SPM possa ser disseminado mais facilmente.

Além das bibliotecas padrões do ambiente de desenvolvimento utilizado, o simulador é composto das seguintes bibliotecas:

- VAR_GLOB.H: Apresenta todas as variáveis globais e definições do sistema;
- ESTRUTUR.H: Contém todas as estruturas de dados usadas;
- PROTOTIP.H: Contém o protótipo de todas as funções e procedimentos;
- ABERTURA.H: Contém as funções da tela de abertura do sistema;
- PROGER.H: Contém as funções gerais que são usadas em outras partes do sistema;
- MENU.H: Contém as funções de manipulação do menu;
- FUN_MENU.H: Contém as funções que são chamadas a partir do menu;
- INICIALI.H: Inicializa as estruturas de dados e variáveis;
- ARVORER.H: Contém as funções de manipulação de árvores e grafos;
- ENLACE.H: Contém as funções de manipulação dos enlaces;
- SAIDA.H: Mostra o valor das variáveis de investigação, usadas em pontos estratégicos do simulador, para coletar resultados;
- DESTINO.H: Contém as funções de controle de grupos *multicast*. Podemos

também dizer que é onde está implantado a maior parte do protocolo IGMP;

- ORIGEM.H: Contém as funções de controle das fontes *multicast*;
- SINALIZA.H: Contém os procedimentos de controle e criação de pacotes de sinalizações;
- UNICAST.H: Contém os procedimentos de controle de transferências *unicast* ;
- START.H: Contém os procedimentos de controle de pacotes de dados e tempo de simulação.
- MOSPF.H: Contém as funções relacionadas ao protocolo MOSPF;
- CBT.H: Contém as funções relacionadas ao protocolo CBT;
- PIM-SM.H: Contém as funções relacionadas ao protocolo PIM-SM;

4.4. PARÂMETROS DE ENTRADA DO SPM

Para que o usuário possa executar uma simulação, é necessário que informe todos os parâmetros solicitados pelo simulador e/ou que sejam obtidos de arquivos previamente alimentados.

A cada execução do simulador é solicitado ao usuário se este deseja ler informações de um arquivo que contenha os parâmetros de entrada. Este arquivo de dados é criado e atualizado após a digitação dos parâmetros de entrada, para que estes possam ser recuperados em execuções futuras.

Esses parâmetros são solicitados a partir das primeiras três opções do menu principal, sendo organizados em:

- Entrada de Dados Gerais;
- Tabela de Rotas; e
- Capacidade dos Enlaces.

4.4.1. ENTRADA DE DADOS GERAIS

Nesta opção são informados os parâmetros gerais para a execução da simulação. A Figura 40 apresenta a tela de interface correspondente à entrada de dados gerais. As opções são descritas a seguir.

Figura 40. Dados gerais – Tela de Entrada

• Número de Roteadores: Este parâmetro define o número de roteadores que participarão da simulação. Cada roteador está ligado a uma ou mais redes. Eles são definidos como "*Designated Router*" de pelo menos uma rede.

• Número Máximo de Grupos: Define o número máximo de grupos que poderemos ter ao mesmo tempo durante a execução da simulação.

OBSERVAÇÃO: Estes dois primeiros parâmetros definem a quantidade de memória necessária para a representação das tabelas de roteamento em cada roteador. De acordo com os protocolos MOSPF e PIM-SM, o número máximo de entradas simultâneas em cada tabela é :

Número de grupos X Número de fontes possíveis

Para que possamos representar isso em termos de estruturas de dados no SPM,

temos:

Número de grupos X Número de fontes possíveis X Número de roteadores

Se definirmos que temos 50 grupos , 10 roteadores e que todos os roteadores podem ser fontes de todos os grupos , teríamos:

50 X 10 X 10 = 5.000 entradas

Como as entradas são controladas em árvores de roteamento, definidas através de estruturas de dados no SPM, podemos ter problemas de memória ao simularmos os protocolos de roteamento *multicast*.

Após várias simulações, concluímos que na atual arquitetura do SPM podemos ter no máximo 1.900 estruturas de dados representando árvores de roteamento. Esta limitação é decorrente da áreas de memória delimitada pelo compilador *Boorland* C, que gerencia apenas 1 Mbyte de memória.

- Número do roteador *Rendezvou Point*: Define o roteador que centralizará as informações *multicast* (*Rendezvou Point*) para os protocolos PIM e CBT. Deve ser especificado o número de um dos roteadores da simulação, ou seja, o número pode variar de 1 até o número total de roteadores que farão parte da simulação.

- Atraso do Roteador: Determina o tempo de processamento dos roteadores para a transferência de um pacote. Na prática é o tempo gasto pelo roteador na análise do pacote e escolha do caminho que ele deve seguir, de acordo com a tabela de rotas. Este valor deve ser expresso em milisegundos.

- Tempo de Simulação: Este valor define a duração da simulação e deve ser expresso em minutos.

- Tamanho do Pacote de Dados: Este parâmetro define o tamanho dos pacotes de dados. O valor deve ser expresso em octetos.

- Tamanho do Pacote de Sinalização: Este parâmetro define o tamanho dos pacotes de sinalização utilizado pelos protocolos a serem simulados. O valor deve ser expresso em octetos. São considerados pacotes de sinalização aqueles que são utilizados pelos protocolos

para troca de informações de topologia e gerência de grupos *multicast*.

• Tamanho do *Buffer* dos Roteadores: Este parâmetro indica o tamanho do *buffer* dos roteadores usados na simulação. O valor deve ser expresso em *Kbytes*.

• Quantidade de Pacotes por Rajada: Este parâmetro define a quantidade de pacotes simultâneos que devem ser produzidos (tamanho do *burst*), ao ser gerado um pacote *multicast*, caracterizando assim, a rajada de um tráfego multimídia.

• Intervalo entre Pulsos: Este parâmetro define o intervalo entre pulsos para a atualização do relógio do simulador. O relógio é incrementado com um valor aleatório entre 0 e aquele especificado em milisegundos.

• Probabilidade de Pacotes *Multicast*: Este parâmetro define o probabilidade de pacotes *multicast* a serem gerados. Por exemplo: de cada 100 pacotes gerados, 20 (valor do parâmetro) serão *multicast* e 80 *unicast*.

• Probabilidade de Novo Grupo: Este valor define a probabilidade de um pacote *multicast* ser gerado para um grupo já existente ou para um novo grupo. Por exemplo: de cada 100 pacotes gerados, 20 (valor do parâmetro) serão para um novo grupo e 80 para um grupo já criado. Este parâmetro fica inativo, no caso de todos os grupos possíveis já terem sido criados (parâmetro definido em Número Máximo de Grupos) ou não existir nenhum grupo definido no momento.

• Probabilidade de mesma fonte *Multicast*: Este parâmetro indica a probabilidade de ser a mesma fonte para um grupo já existente. Por exemplo: de cada 100 pacotes gerados para um determinado grupo, 80 (valor do parâmetro) serão de uma fonte já existente no grupo e 20, de uma fonte que será criada. Este parâmetro ficará inativo caso todas as fontes possíveis de um grupo tenham sido criadas ou não existam fontes criadas para o grupo.

• Tempo Entre *Query*: Este parâmetro define o intervalo de tempo em que o roteador espera para solicitar informações sobre associações para um determinado grupo, à rede que o elegeu como "*Designated Router*". O valor deve ser especificado em segundos, de acordo com a especificação do protocolo IGMP.

• Atraso de um *Respond*: Tempo que o roteador espera por uma resposta de uma

solicitação de associações ao par (Grupos, Fonte). Caso não seja enviada uma resposta em tempo hábil, inicia-se o processo de "poda". O valor deve ser especificado em milisegundos, de acordo com a especificação do protocolo IGMP.

4.4.2. TABELA DE ROTAS

A tabela de rotas deve representar a topologia da rede que será simulada. A primeira coluna representa os roteadores de origem, e a primeira linha representa os roteadores de destino. A intercessão de uma linha com uma coluna define o próximo caminho que o pacote irá percorrer para ser entregue ao destino. Quando a intercessão de uma linha com uma coluna é representada pelo número -1, significa que o pacote atingiu o seu destino.

Figura 41. Tabela de Rotas – Tela de entrada

4.4.3. CAPACIDADE DOS ENLACES

A capacidade dos enlaces deve refletir a quantidade de *bits* que podem trafegar de um roteador para outro em 1 segundo. Os enlaces foram definidos como sendo *full-duplex* e devem ser expressos em *Kbits*/segundo.

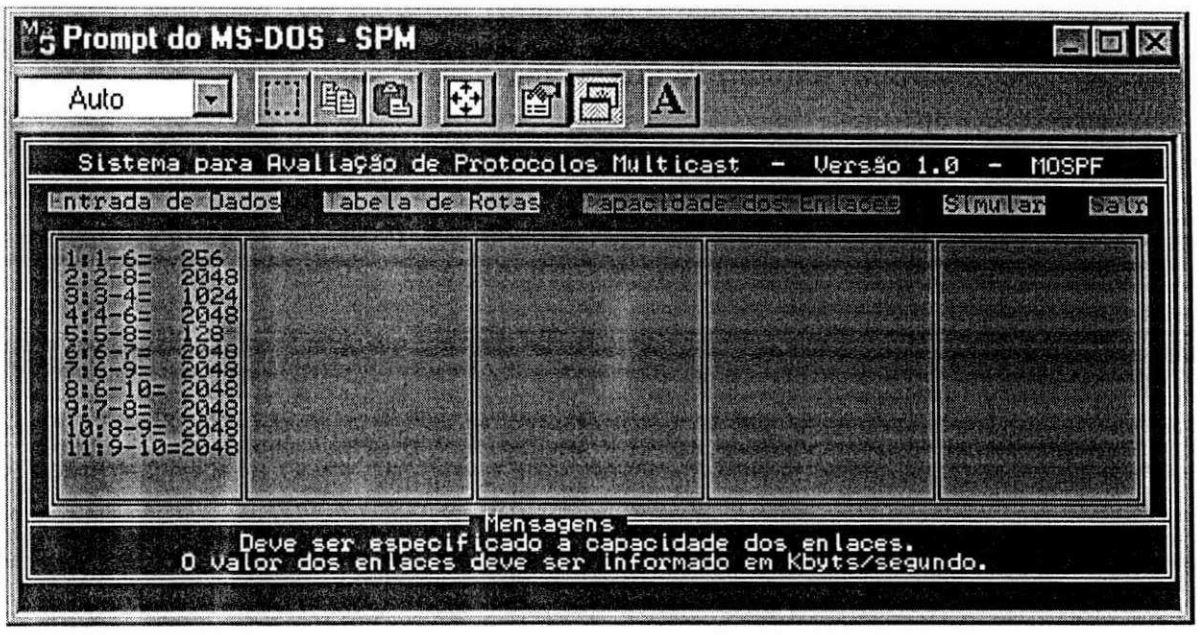


Figura 42. Capacidade dos Enlaces – Tela de Entrada

4.5. Resultados da Simulação

Os resultados da simulação são obtidos através de variáveis, colocadas estrategicamente em determinadas funções do simulador, chamadas de pontos de investigações. Elas são divididas em três categorias:

- Controle de tráfego;
- Controle do tamanho máximo das tabelas de roteamento;
- Controle de ocupação dos enlaces.

Apresentamos a seguir as medidas de desempenho definidas para cada categoria referenciada.

4.5.1. Controle de Tráfego

- Tempo da Simulação: Tempo de duração da simulação em milisegundos;
- Pacotes de Sinalização: É a soma de todos os pacotes usados para gerência e controle das funções envolvidas na transação de dados *multicast*;

• Pacotes *Multicast*: É a quantidade de pacotes gerados do tipo *multicast*;

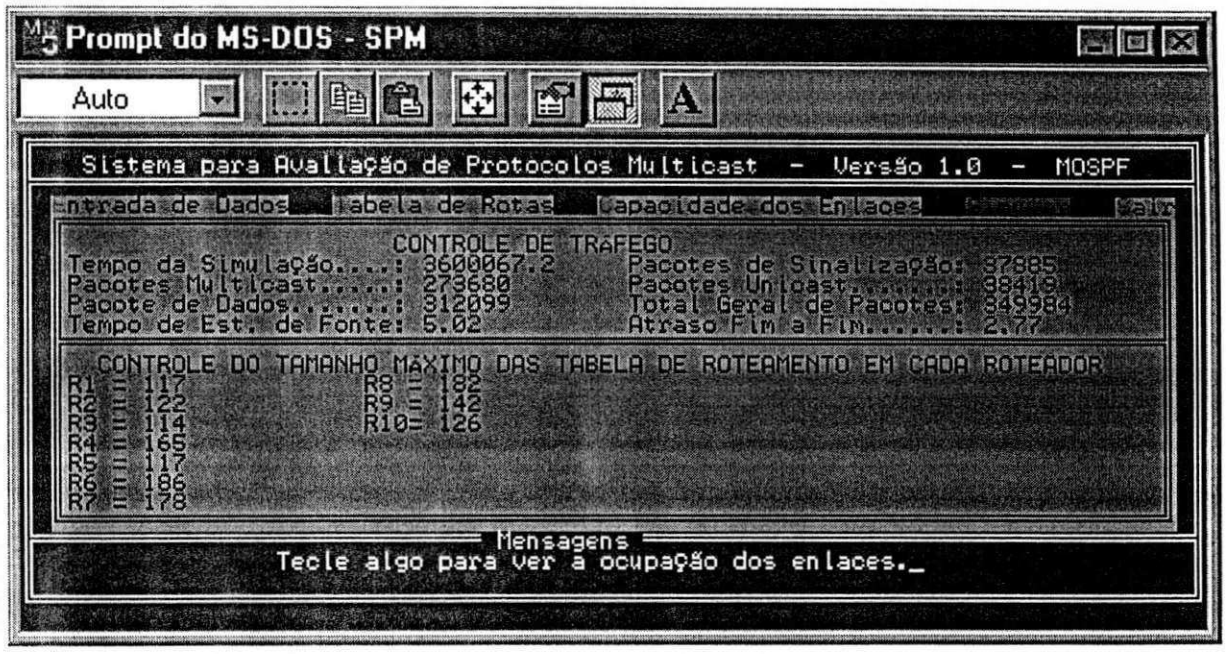


Figura 43. Controle de Tráfego

• Pacotes *Unicast*: É a quantidade de pacotes gerados do tipo *unicast*;

• Pacote de Dados: É a soma de pacotes *unicast* e *multicast*;

• Total Geral de Pacotes: É a soma dos pacotes de dados com os pacotes de sinalização.

•Tempo Médio de Estabelecimento de Fonte: É o tempo médio decorrido para o estabelecimento de uma fonte em milisegundos, considerando o tráfego no momento. O cálculo é feito da seguinte forma:

$$TEF = \frac{\sum \text{Tempo decorrido para a criação de cada Fonte}}{\sum \text{de Fontes}}$$

• Atraso Médio Fim a Fim: É o tempo médio decorrido para a entrega de um pacote *multicast* a todos os membros de um determinado grupo, em milisegundos, considerando o tráfego no momento. O calculo é feito da seguinte forma:

$$AFF = \frac{\sum \text{ do tempo decorrido para a entrega de todos os pacote}}{\sum \text{ dos Pacotes gerados}}$$

4.5.2. CONTROLE DO TAMANHO MÁXIMO DAS TABELAS DE ROTEAMENTO

• Apresenta para cada roteador (R1, R2,...,Rn) o tamanho máximo que cada tabela de roteamento atingiu durante a simulação. Estes valores são importantes para determinar a utilização de memória, velocidade de processamento e performance do Sistema Operacional.

4.5.3. CONTROLE DA OCUPAÇÃO DOS ENLACES.

• Enlaces: Especifica o enlace que foi investigado;

• Quantidade de Pacotes de Dados: É a quantidade de pacotes de dados que trafegaram pelo enlace;

• Quantidade de Pacotes de Sinalização: É a quantidade de pacotes de sinalização que trafegaram pelo enlace;

• *Overflow*: É a quantidade de pacotes descartados, devido à utilização completa dos *buffers* relacionados com o enlace;

• Quantidade Total: É a quantidade de pacotes de dados, mais a quantidade de pacotes de sinalização, mais a quantidade de *Overflow*.

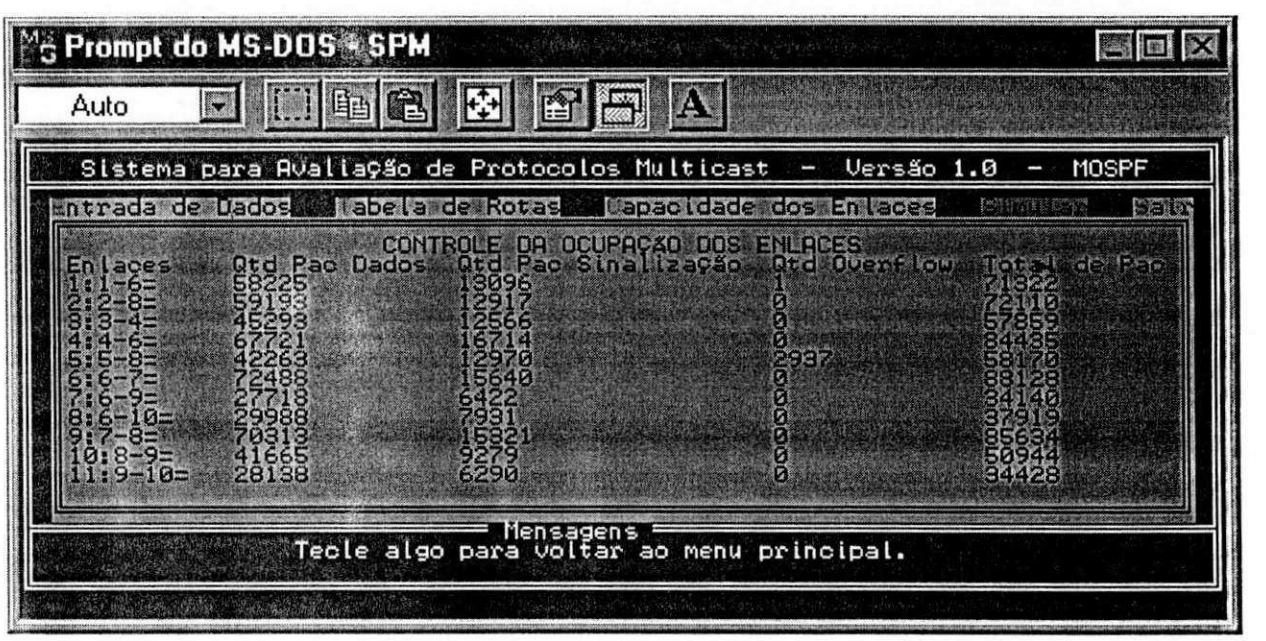


Figura 44. Controle da ocupação dos enlaces

CAPÍTULO 5

SIMULAÇÃO DE PROTOCOLOS *MULTICAST* – ESTUDO DE CASO: RNP

A implementação *multicast* pode ser desenvolvida sobre várias tecnologias de rede, entre elas podemos destacar *Frame Relay*, SMDS (*Switched Multimegabit Data Service*), ATM (*Asynchronous Transfer Mode*), ISDN (*Integrated Services Digital Network*), *Ethernet*, *Token Ring*, entre outras. Todas estas tecnologias usualmente habilitam a família de protocolos TCP/IP para trabalhar em conjunto com os seus protocolos nativos.

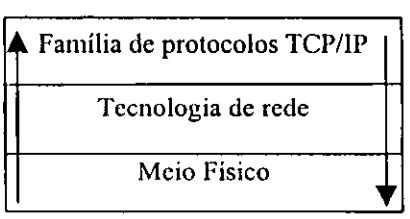


Figura 45. Interoperabilidade entre tecnologias de Rede

Como acontece em todas as propostas de simulações, devemos definir um escopo de atuação, que nos sirva como guia para a entrada dos parâmetros do modelo a ser simulado.

Para que possamos apresentar resultados sobre a avaliação dos protocolos *multicast*, definimos como escopo de atuação o *backbone* da RNP, do qual, consideramos sua topologia, a taxa de tráfego atual gerada na rede, as velocidades de comunicação entre os roteadores que compõem o *backbone* e as funcionalidades dos protocolos utilizados atualmente, além de incorporarmos as funcionalidades dos protocolos *multicast*, os quais são de interesse para as nossas avaliações.

Neste capítulo apresentamos a descrição e funcionamento da RNP, os modelos dos protocolos de roteamento *multicast* que foram acoplados ao ambiente real da RNP, e a definição dos valores atribuídos aos parâmetros de entrada nas simulações realizadas.

5.1. DESCRIÇÃO DA RNP

"A RNP é uma iniciativa do Ministério da Ciência e Tecnologia, cuja missão básica é planejar e conduzir ações que assegurem a implantação e evolução de redes *Internet* no Brasil. A audiência central dos serviços da RNP é a comunidade de educação, pesquisa e desenvolvimento científico e tecnológico, e gestão governamental nessas áreas. Não obstante, visando impulsionar a efetiva utilização de Internet em todo o país, a RNP oferece acesso as instituições de qualquer natureza, sem prejuízo de sua missão básica, em praticamente todas as capitais do país" [RNP 98].

A RNP oferece serviços de redes *Internet* através de uma malha de conexões dedicadas interligando hoje praticamente todas as capitais do país, compondo um *backbone* nacional.

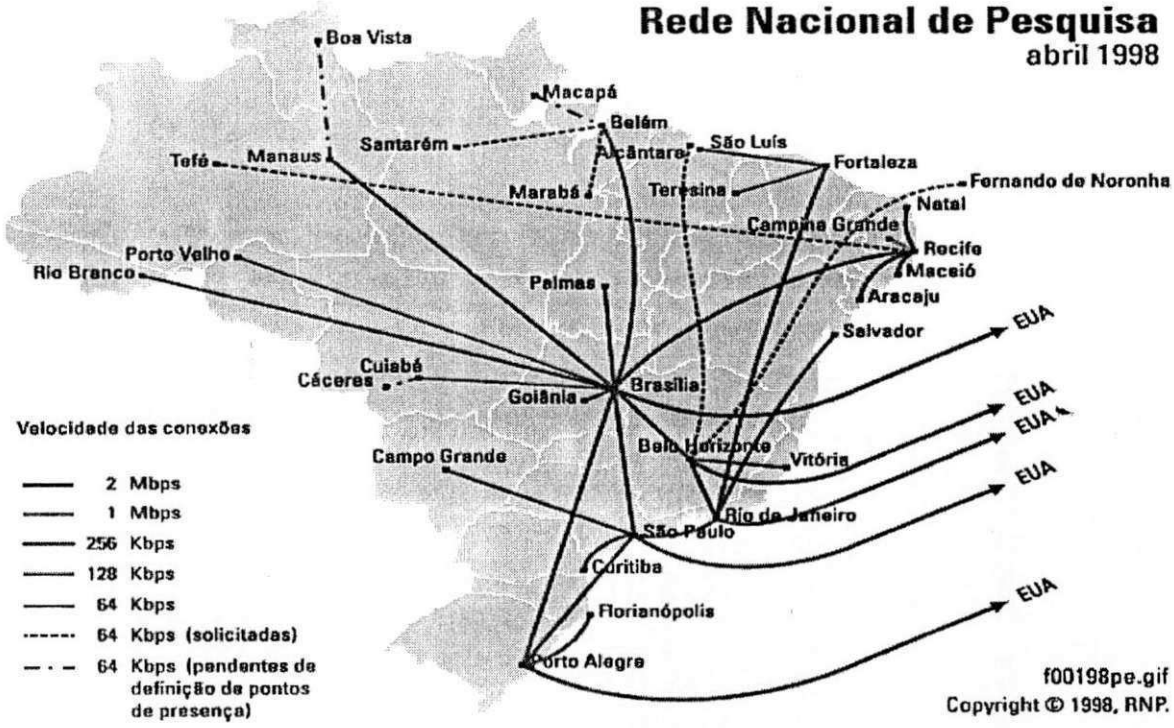


Figura 46. *Backbone* RNP

O meio físico que interliga os roteadores no *backbone* usa a tecnologia de transmissão via rádio, com o protocolo HDLC (*High-Livel Data Link Protocol*) [COMER 91] a nível de enlace de dados , permitindo assim, conexões ponto-a-ponto síncronas.

A camada superior é implementada com a família de protocolos TCP/IP, desta forma possibilitando a implementação *multicast*, como vimos nos capítulos 2 e 3.

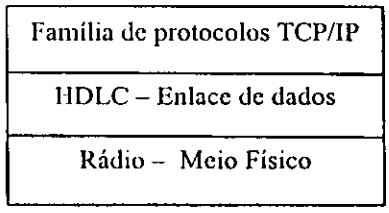


Figura 47. Tecnologia de rede usada nos roteadores do *backbone* da RNP

O tráfego que passa pelo *backbone* da RNP é gerado pelos usuários das redes que se ligam ao PoP (Ponto de Presença) da RNP em cada estado. Estas redes podem ser de diferentes tecnologias, como apresentado no primeiro parágrafo deste capítulo.

5.2. ESPECIFICAÇÃO DO MODELO DOS PROTOCOLOS *MULTICAST*

Os protocolos *multicast* utilizados para integrar o escopo do *backbone* da RNP, foram escolhidos de acordo com as necessidades de avaliar as diferentes técnicas de construção de árvores *multicast*.

Como definido no capítulo 3 existem três técnicas para a construção de árvores: a árvore SPT, a árvore RP e uma técnica híbrida, utilizando inicialmente a árvore RP e após a árvore SPT.

A motivação pela escolha de um dos protocolos que utilizam a árvore SPT (DVMRP, MOSPF e PIM-DM) baseou-se em [RFC 1585], onde o autor sugere a análise do **MOSPF** em outros cenários, a fim de avaliar o possível comportamento do protocolo, com parâmetros diferentes daqueles de sua análise.

Os outros dois protocolos escolhidos foram o **PIM-SM** (híbrido) e o **CBT** (árvore RP), por implementarem técnicas de construção de árvores que não são comumente utilizadas em outros protocolos. Apresentamos a seguir a modelagem do MOSPF, PIM-SM e CBT.

5.2.1. MODELAGEM DO MOSPF

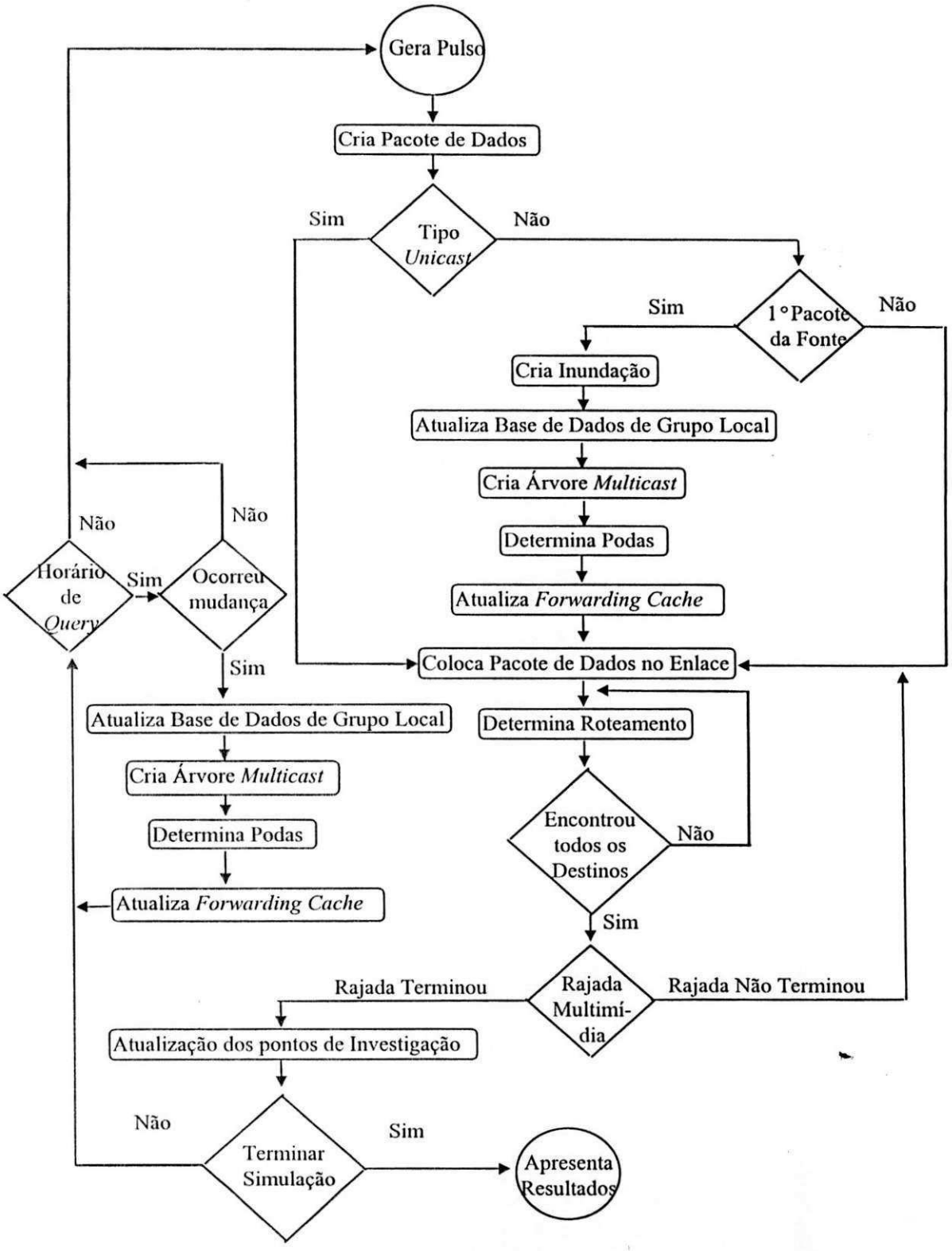


Figura 48. Modelagem do MOSPF

• O inicio da simulação ocorre a partir do primeiro pulso gerado, dando início à criação da estrutura do pacote de dados e incremento do relógio da simulação;

• O próximo passo é descobrir se o pacote é *unicast* ou *multicast*:

→ Caso seja *multicast* e de uma nova fonte:

- É executada uma inundação, para divulgação da existência da nova fonte;
- Em seguida, são geradas as perguntas IGMP pelos "Roteadores Designados", para saber se existem *hosts* na rede local interessados em juntar-se à nova fonte, e assim, atualizadas as "Bases de Dados de Grupo Local" de acordo com as respostas obtidas;
- Neste instante, começa a ser criada a árvore SPT entre a fonte e todos os roteadores do domínio;
- Em seguida, é executado o processo de poda dos galhos que não serão utilizados, baseados na "Base de Dados de Grupo Local";
- Com o resultado das árvores SPT após a poda, são criadas entradas no *Forwarding Cache* e dispensados os recursos envolvidos com as árvores SPT;
- A partir deste ponto, segue o fluxo normal com os demais passos, explicados abaixo.

• O pacote é colocado no enlace e segue seu caminho através da árvore de roteamento até alcançar o seu destino;

• Caso seja um pacote *multicast*, aplica-se a concepção de que foi gerada uma rajada de dados multimídia, sendo assim, coloca-se todos os pacotes da rajada nos enlaces;

• Em seguida, é feita a atualização dos pontos de investigação;

• O próximo passo é testar se o tempo decorrido é maior que o tempo determinado para a simulação. Em caso afirmativo, termina a execução, sendo apresentado os resultados obtidos nos pontos de investigação;

• Caso não termine a execução da simulação, é testado o tempo para uma nova pergunta do "*Designated Router*" à rede, para descobrir se existem mudanças na "Base de Dados de Grupos Local", seguindo os mesmos passos descritos anteriormente, para o caso de ser *Multicast* e de uma Nova Fonte;

• Por fim, é gerado um novo pulso e continua a execução da simulação.

5.2.2. MODELAGEM DO PIM-SM

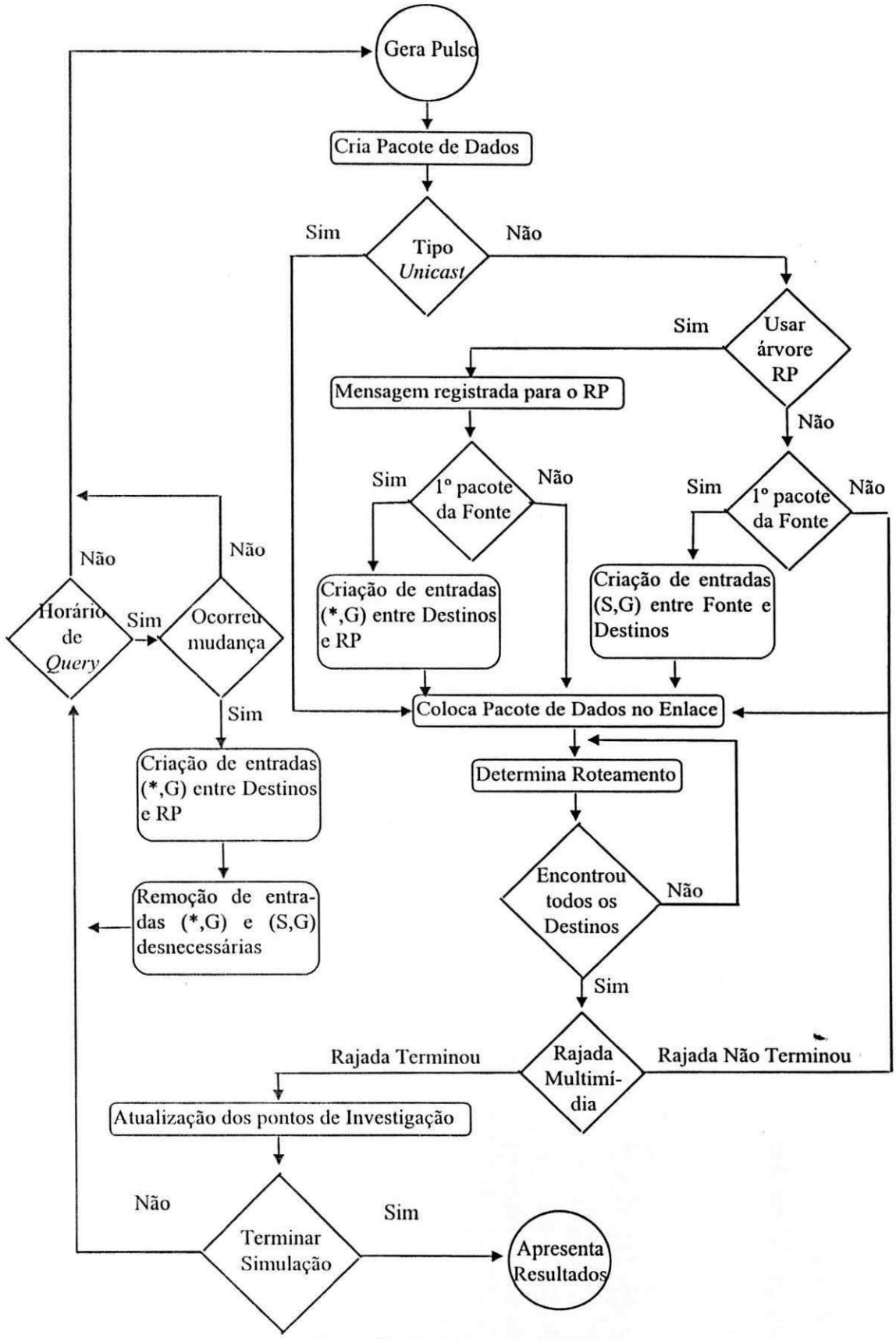


Figura 49. Modelagem do PIM-SM

Descrevemos aqui somente as particularidades da modelagem do protocolo PIM-SM.

- Caso os pacotes sejam roteados pela árvore RP:

  → As mensagens são registradas e enviadas na forma *unicast* para o RP;

  → Se for o primeiro pacote gerado pela fonte:

    - Os roteadores que possuem membros que desejam associar-se à nova fonte, enviam uma mensagem que passa de roteador a roteador até chegar ao RP, criando a entrada (*,G) em todos eles.

  → O RP desencapsula o pacote que veio em uma mensagem registrada da fonte e o coloca nos enlaces que fazem parte da árvore RP.

- Caso os pacotes sejam roteados pela árvore SPT:

  → Se for o primeiro pacote gerado pela fonte para a árvore SPT:

    - Os roteadores que possuem membros que podem associar-se à fonte via árvore SPT, enviam uma mensagem que passa de roteador a roteador até chegar à fonte, criando a entrada (S,G) em todos eles.

  → A fonte coloca o pacote nos enlaces que fazem parte da árvore SPT.

- Caso seja Hora de *Query*:

  → Se tiver ocorrido mudanças:

    - Cria-se as entradas (*,G) nos roteadores entre os novos destinos e o RP;

    - Remove-se as entradas (*,G) e (S,G) desnecessárias .

5.2.3. MODELAGEM DO CBT

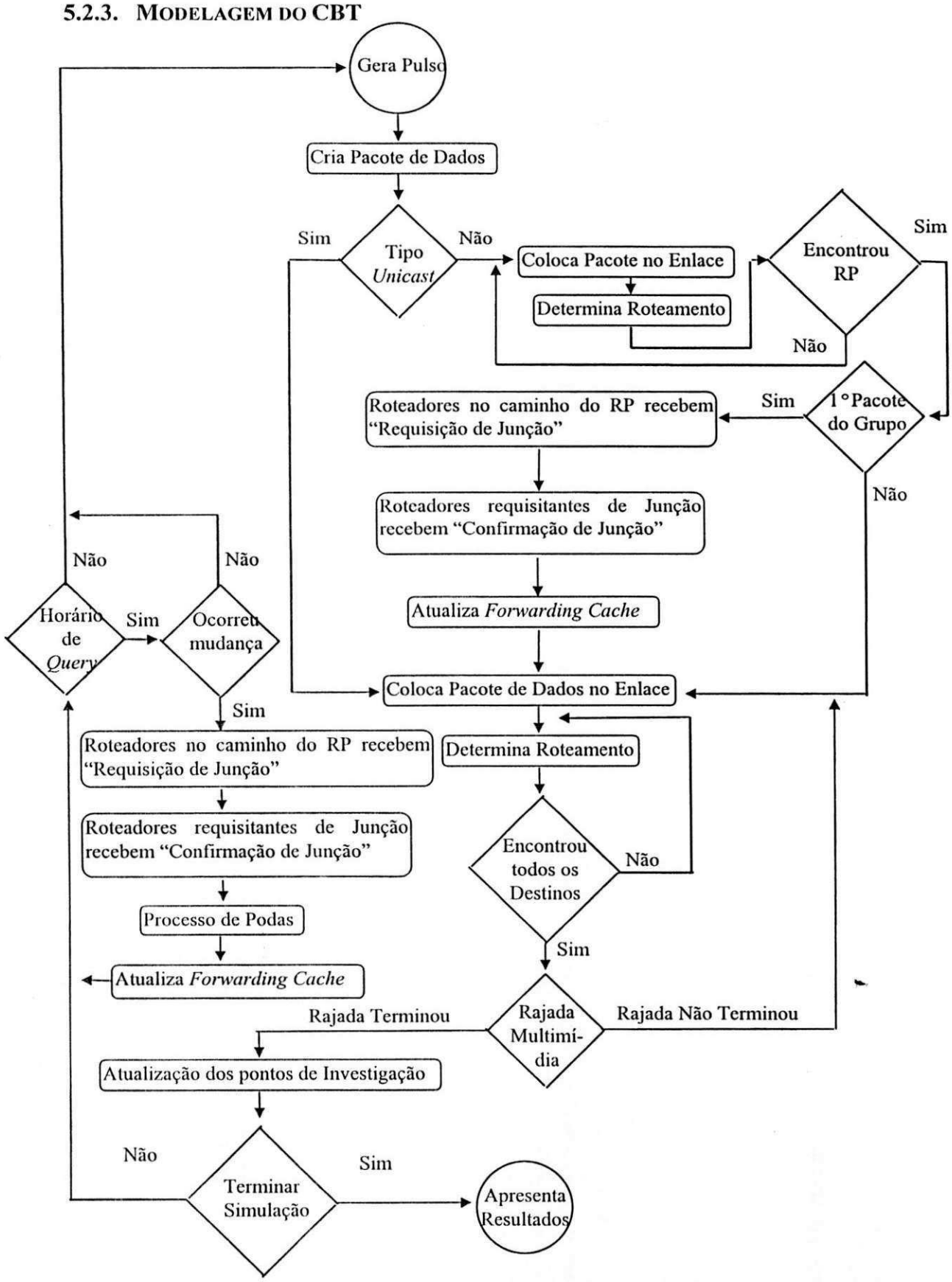


Figura 50.Fluxo do modelo CBT absorvido pelo SPM

Descreveremos aqui somente as particularidades da modelagem do protocolo CBT.

- Os pacotes são roteados das fontes até o RP;
- Caso seja o primeiro pacote de um grupo *multicast*:

  → Os roteadores que possuem membros que desejam se associar ao novo grupo enviam mensagens de "Requisição de Junção" em direção ao RP;

  → Os roteadores que solicitaram junção recebem mensagens de "Confirmação de Junção", diretamente do RP ou de roteadores no caminho do RP;

  → As Memórias de Entrega dos roteadores que fazem parte da árvore RP são atualizadas.

- Caso seja Hora de *Query*:

  → Se tiver ocorrido mudanças:

  - Os roteadores que possuem membros que desejam se associar a algum grupo estabelecido enviam mensagens de "Requisição de Junção" em direção ao RP;
  - Os roteadores que solicitaram junção recebem mensagens de "Confirmação de Junção", diretamente do RP ou de roteadores no caminho do RP;
  - Os roteadores que não possuem mais interesse em receber pacotes de um determinado grupo enviam "Notificações de Saída" ao seu roteador pai, na árvore RP, indicando poda;
  - As Memórias de Entrega dos roteadores que fazem parte da árvore RP são atualizadas.

5.3. Definição dos Parâmetros de Entrada para o SPM

Neste ponto, enfatizamos os valores colocados nos parâmetros de entrada do SPM, definindo por completo o nosso escopo de simulação. Todos os parâmetros apresentados aqui foram especificados na seções 4.4.

☞ **Número de Roteadores:** Escolhemos 10 *sites* no *backbone* da RNP para a execução de nossas simulações. Os roteadores são definidos de acordo com a seguinte nomenclatura:

1- Manaus (MNS)	2- Fortaleza (FLA)
3- Campina Grande (CGE)	4- Recife (RCE)
5- Salvador (SSA)	6- Brasília (BSA)
7- Belo Horizonte (BHE)	8- Rio de Janeiro (RJO)
9- São Paulo (SPO)	10- Porto Alegre (POA)

☞ **Número máximo de Grupos simultâneos:** Escolhemos 19 por ser o número máximo de grupos suportados pelo simulador SPM na atual arquitetura, como mencionado na seção 4.4.1.

☞ **Roteador *Rendezvou Point*:** Escolhemos o mesmo RP nos protocolos PIM-SM e CBT para possibilitar a avaliação com a mesma parametrização. Além disso, usamos o RP fixo, para facilitar a implementação de ambos os modelos. A escolha de Brasília (6) como RP, foi em função do seu número de conexões com outros *sites* e posição do roteador no cenário da RNP [ZEGU 95].

☞ **Atraso do Roteador:** Definimos um atraso de 1 milisegundo para a manipulação de um pacote, ou seja, o tempo gasto para o roteador decidir em que enlace ou em quais enlaces deve colocar o pacote, de acordo com sua tabela de rotas.

☞ **Tempo de simulação:** Escolhemos 1 hora por ser um tempo razoável de simulação e por não acarretar impactos na obtenção dos resultados.

☞ **Tamanho dos Pacotes de Dados:** Definimos um tamanho de 512 octetos (*bytes*), por ser um valor utilizado em algumas aplicações. Entretanto este valor tem uma variação de aplicação para aplicação, além de variar de acordo com a quantidade de informações que se quer transferir [NORO 94B].

☞ **Tamanho dos Pacotes de Sinalização:** Definimos um tamanho de 128 octetos (*bytes*), por ser um tamanho usual das mensagens que os protocolos *multicast* usam para desempenhar suas funções. Entretanto podem ocorrer variações deste tamanho, de acordo

com a quantidade de informações que um roteador tem que transferir para seus vizinhos [RFC 2201] [CAIN 97].

☞ **Tamanho do *Buffer* dos Roteadores**: Este parâmetro pode ser configurado pelo gerente da rede de acordo com as necessidades de tráfego que passam pelo roteador. Em nossas simulações utilizamos o valor de 5 Kbyts, o que dá a possibilidade de armazenar até 10 pacotes de dados simultaneamente.

☞ **Quantidade de Pacotes por Rajada**: Este parâmetro pode variar muito, pois representa o tamanho de uma rajada multimídia, que está relacionada com as ações executadas em um determinado momento em uma aplicação de vídeo conferência, por exemplo. Definimos o valor de 10 pacotes de dados, o que dá uma quantidade de 5.120 *bytes*.

☞ **Intervalos entre Pulsos**: Definimos este parâmetro com o valor de 150, para que tenhamos uma quantidade de aproximadamente 30.000 *bits/s* saindo de cada roteador, de acordo com estatísticas do PoP da RNP de Porto Alegre [RNP 98]. Como temos 10 roteadores em nossas simulações, são gerados 300.000 *bits/s* aproximadamente.

☞ **Probabilidade de Pacotes *Multicast***: Este parâmetro foi escolhido de acordo com o trabalho de [HARR 95], o qual define 20% do tráfego gerado como sendo *multicast*.

☞ **Probabilidade de Novo Grupo**: Este parâmetro foi definido de acordo com o trabalho de [HARR 95], o qual define 20% de probabilidade de ser gerado um pacote *multicast* para um novo grupo.

☞ **Probabilidade de Mesma Fonte**: Este parâmetro também foi escolhido de acordo com o trabalho de [HARR 95], o qual define 80% de probabilidade de ser gerado um pacote *multicast* que sairá de uma fonte já existente.

☞ **Tempo entre *Query***: Este parâmetro foi escolhido de acordo com o padrão do protocolo IGMP, que define que uma mensagem de "*Query*" deve ser gerada a cada 60 segundos [RFC 1112].

☞ **Tempo entre *Respond***: Este parâmetro foi definido como 40 milisegundos em função de ser um valor válido pela definição do protocolo IGMP [RFC 1112].

☞ **Capacidade dos Enlaces**: Estes parâmetro foram definidos em função da capacidade dos enlaces do *backbone* da RNP, como mostra a Figura 51.

☞ **Tabela de Rotas**: A tabela de rotas (Figura 41) foi montada de acordo com a topologia da rede mostrada na Figura 51, definindo os melhores caminhos de um ponto a outro.

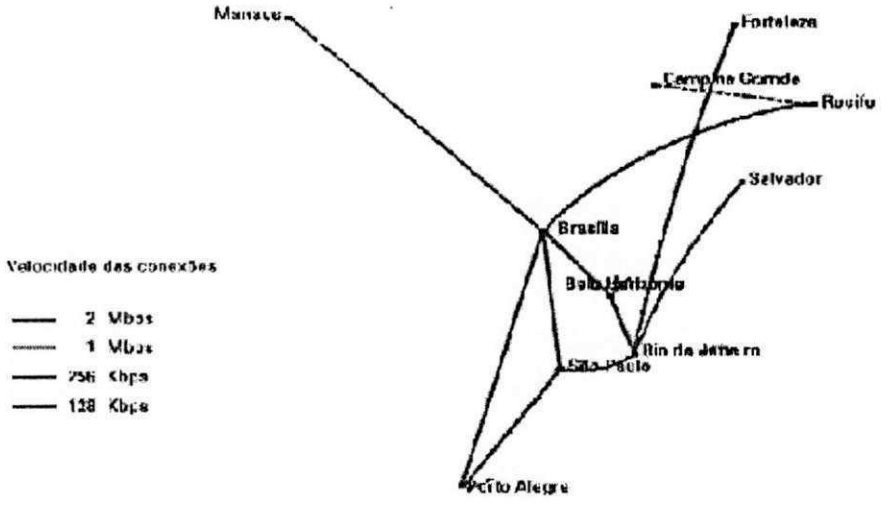
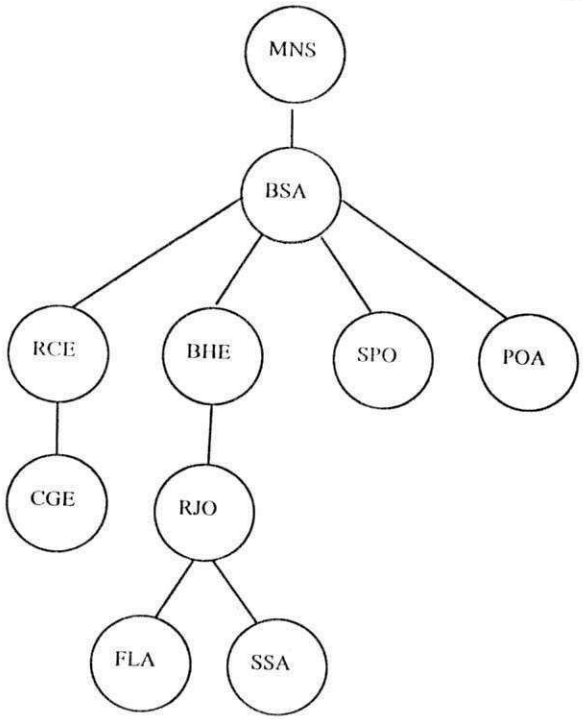
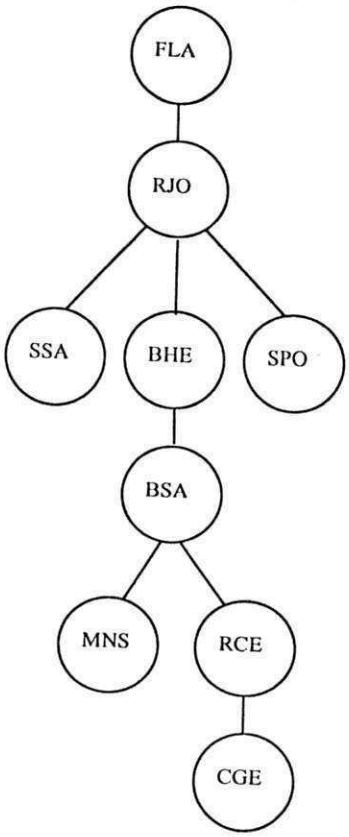


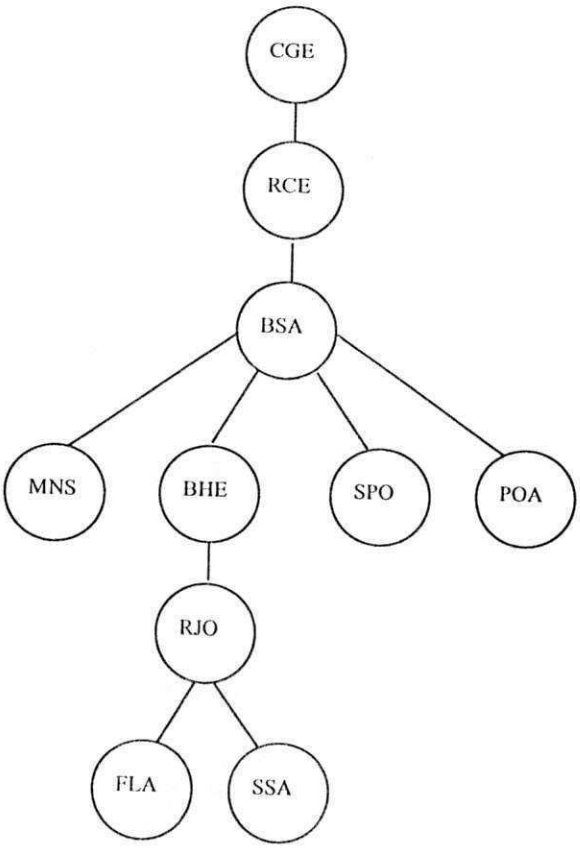
Figura 51. Topologia do estudo de caso usado nas simulações

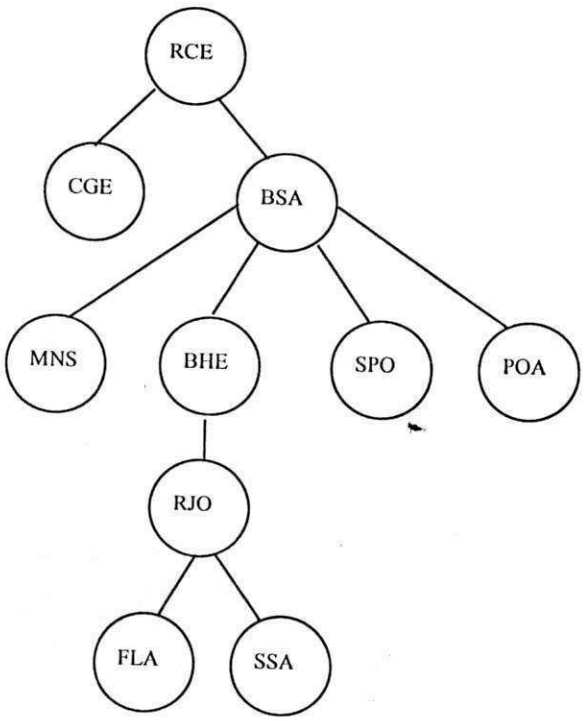
Para facilitar a compreensão das ligações ocasionadas pela tabela de rotas, apresentamos as árvores SPT entre uma fonte que está ligada diretamente ao roteador raiz até os demais roteadores com possíveis destinos. As árvores são construídas a partir da tabela de rotas apresentada na seção 4.3.2, de acordo com o cenário da RNP, indicando os caminhos que o roteador que se encontra na raiz das árvores usa para atingir os demais roteadores.

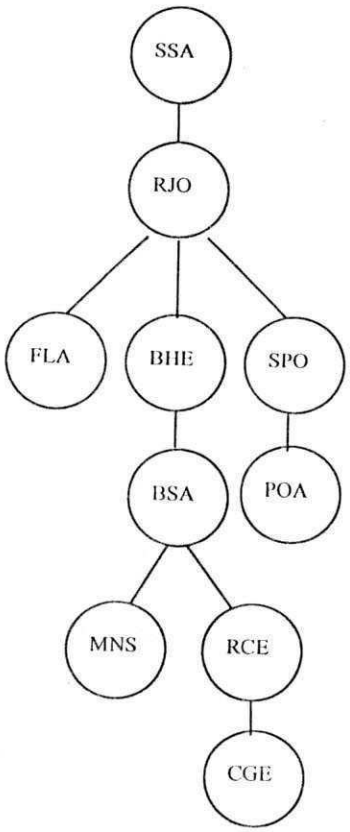
A Figura 52 apresenta as árvores SPT, de acordo com o cenário da RNP. Durante a simulação, estas árvores sofrem "podas" e "reenxertos" de acordo com a existência ou não de membros de um par (Grupo, Fonte). A árvore que possui o roteador de Brasília (BSA) como raiz é a árvore RP.

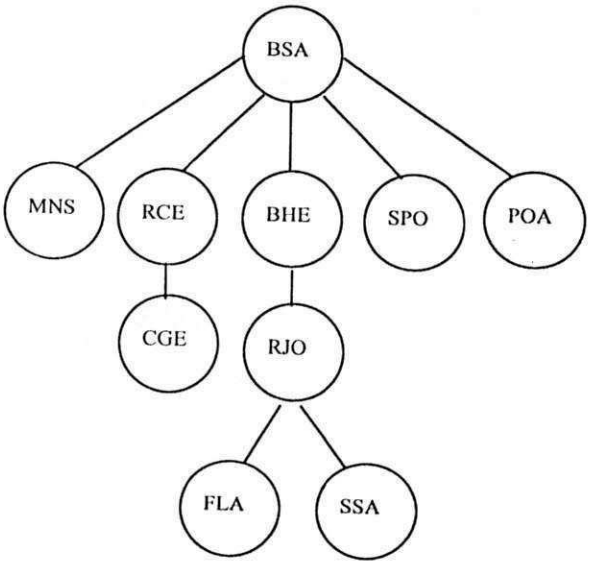

MNS
BSA
RCE
BHE
SPO
POA
CGE
RJO
FLA
SSA

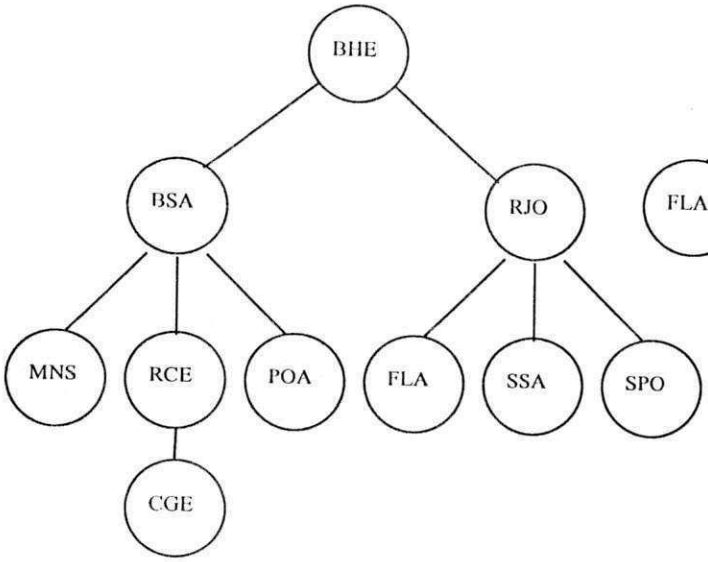

FLA
RJO
SSA
BHE
SPO
BSA
MNS
RCE
CGE

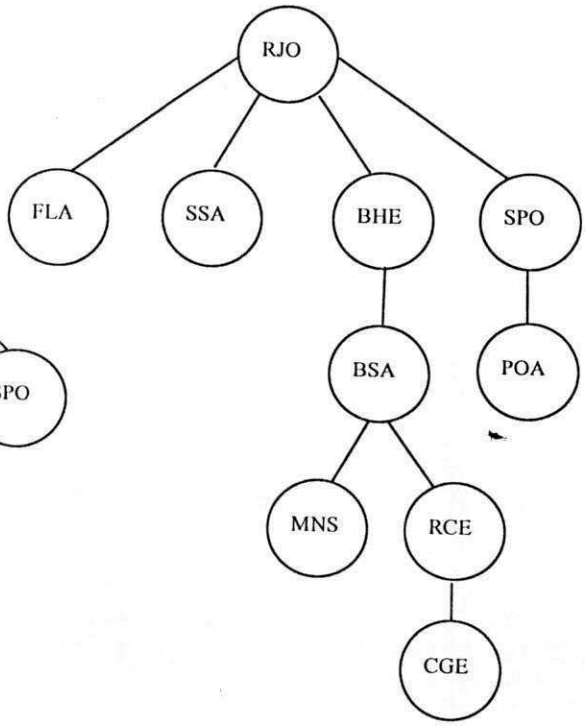

CGE
RCE
BSA
MNS
BHE
SPO
POA
RJO
FLA
SSA


RCE
CGE
BSA
MNS
BHE
SPO
POA
RJO
FLA
SSA


SSA
RJO
FLA
BHE
SPO
BSA
POA
MNS
RCE
CGE


BSA
MNS
RCE
BHE
SPO
POA
CGE
RJO
FLA
SSA


BHE
BSA
RJO
MNS
RCE
POA
FLA
SSA
SPO
CGE


RJO
FLA
SSA
BHE
SPO
BSA
POA
MNS
RCE
CGE

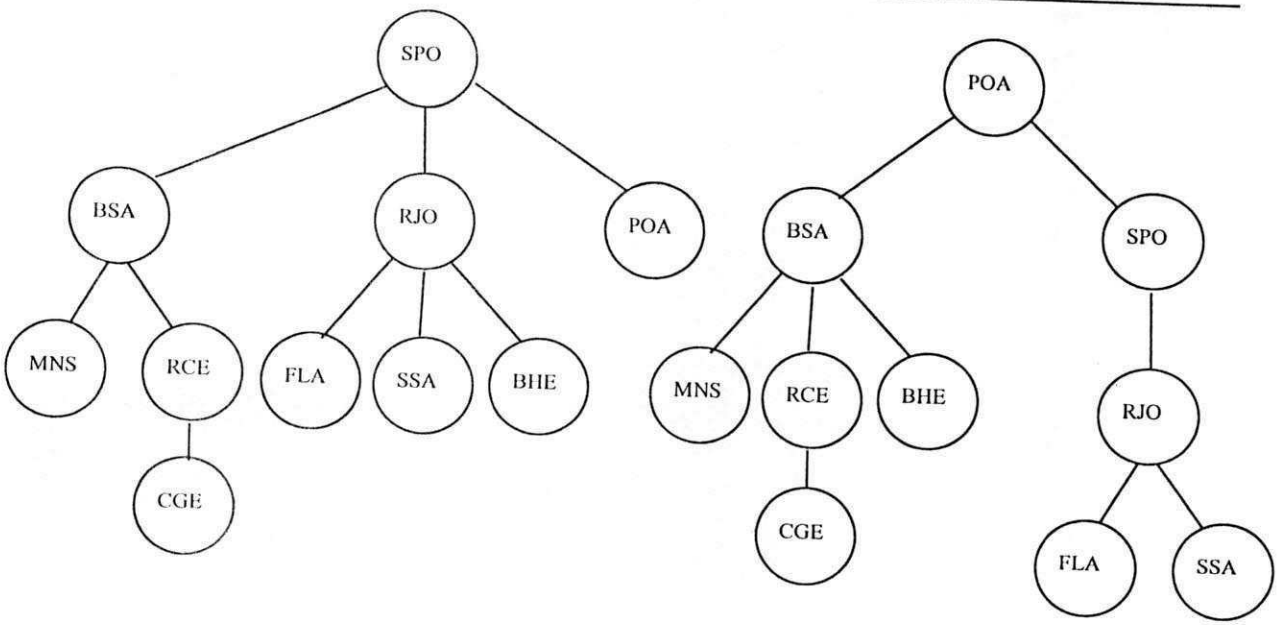


Figura 52. Árvores SPT de acordo com o cenário da RNP

CAPÍTULO 6

AVALIAÇÃO DOS RESULTADOS DAS SIMULAÇÕES

Os resultados obtidos através de simulações nos permitem inferir sobre o comportamento de um determinado sistema em função dos parâmetros de entrada determinados para a execução de um modelo. Em muitos casos os resultados são fundamentais para a definição das exigências para a implantação, descoberta de erros, verificação de futuras necessidades, estimativas de custos e avaliação da performance do sistema.

Com esse intuito, avaliamos os protocolos MOSPF, PIM-SM e CBT, a fim de colaborarmos com os estudos de performance dos protocolos *multicast*, no que diz respeito à geração de tráfego, controle de atrasos médios, comportamento das tabelas de roteamento, ocupação e sobrecarga dos enlaces.

Todos os resultados que serão apresentados são valores médios calculados a partir de amostragens obtidas após a execução de 35 simulações para cada protocolo, com os mesmos parâmetros de entrada (seção 5.3), utilizando o simulador SPM.

Os enlaces não apresentam problemas (falhas, por exemplo) no decorrer das simulações, consequentemente não ocorrem mudanças na topologia da rede construída pela tabela de rotas informada ao simulador como parâmetro de entrada. Os grupos *multicast* são gerados dinamicamente e atualizados em intervalos de tempo definidos para o protocolo IGMP.

Ao longo da discussão, nos referenciaremos a outros autores que desenvolveram

trabalhos teóricos, práticos, ou usaram simulações para avaliar os protocolos *multicast*, permitindo que possamos validar nossos resultados em função das características, funcionalidades e aspectos operacionais dos protocolos, de acordo com as considerações apresentadas na seção 3.7.

A apresentação dos resultados será feita através de tabelas que mostram os resultados numéricos e possibilitam a construção de gráficos, como veremos no restante do capítulo.

6.1. Tráfego de cada Protocolo Investigado

Essas medidas, apresentadas na Tabela 2 e na Figura 53, nos permitem identificar o comportamento do tráfego *unicast*, *multicast* e de sinalização que são gerados, permitindo com isso, identificar o custo que cada protocolo requer para entregar os pacotes a todos os destinos de um par (Grupo, Fonte) e a quantidade de pacotes de sinalização necessária para o funcionamento dos mesmos.

Protocolos	Sinalização	*Multicast*	*Unicast*	Dados	Total Geral
MOSPF	38.090	276.770	38.165	314.935	353.025
PIM-SM	36.860	277.510	38.247	315.757	352.617
CBT	2.619	350.430	38.355	388.785	391.476

Tabela 2. Tráfego gerado

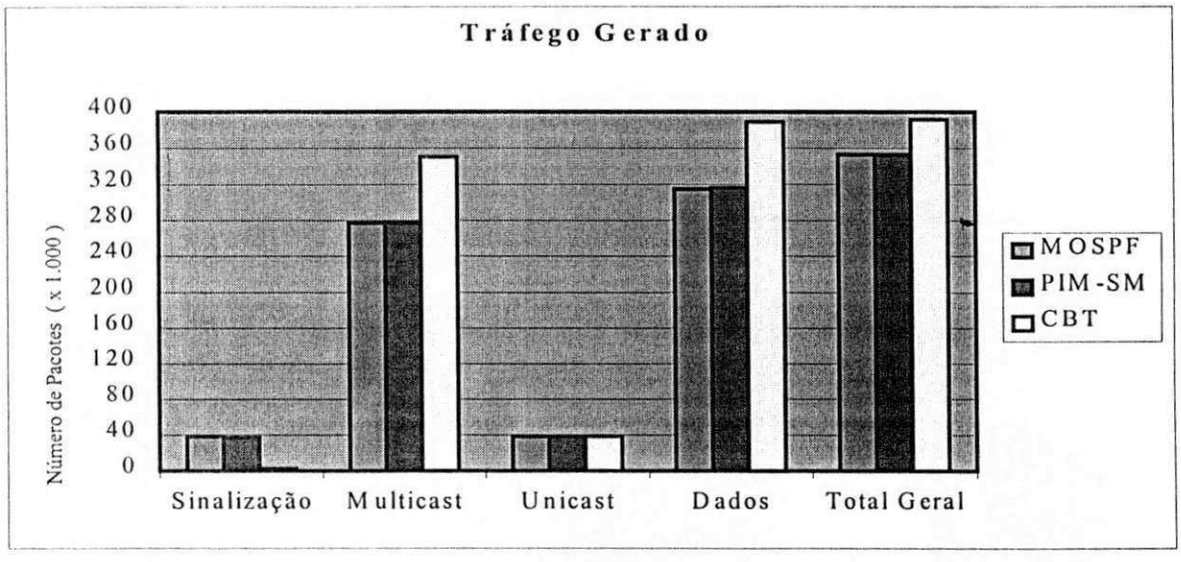


Figura 53.Tráfego Gerado

Podemos observar que o número de pacotes *unicast* é semelhante em todos os protocolos, como já estávamos esperando. Por outro lado, as diferenças apresentadas no tráfego *multicast*, estão relacionadas com a forma de contagem do tráfego gerado, já que os pacotes são calculados de acordo com as replicações sofridas para atingir todos os destinos de um grupo, como mostra a Figura 54.

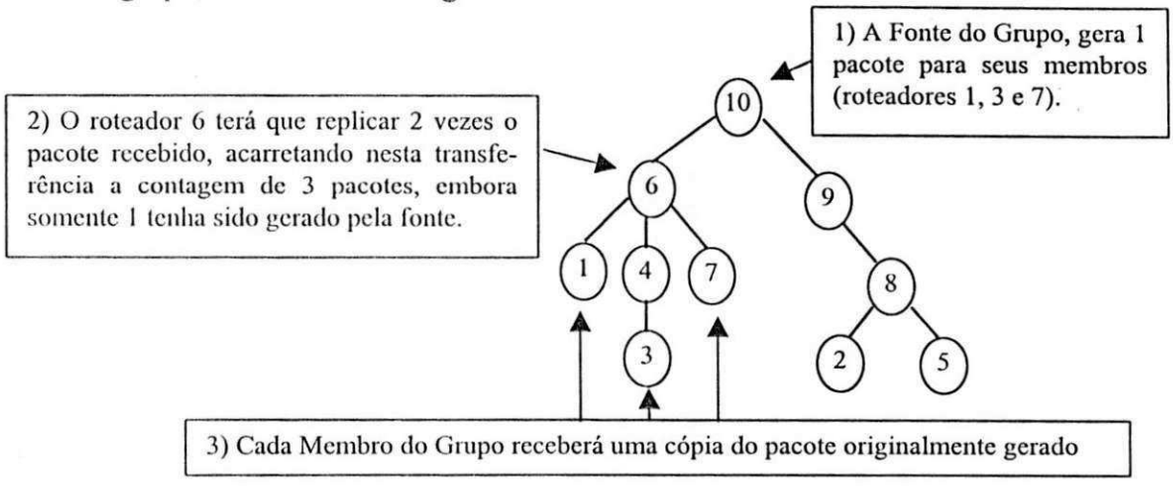


Figura 54. Forma de contagem dos pacotes *multicast*

Desta forma, o CBT apresenta maior tráfego *Multicast* devido ao fato de manter sempre a árvore de entrega pelo *Rendezvou Point*, não utilizando assim, na maioria das vezes, o menor caminho entre uma fonte e os destinos, o que exige maior número de replicações dos pacotes. Esta explicação também é valida para a pequena diferença apresentada entre o MOSPF e PIM-SM, pois o primeiro sempre utiliza o menor caminho, enquanto que o segundo, só o utiliza depois de uma certa quantidades de pacotes gerados.

Assim, como foi encontrado em [FARR 97], o número de pacotes de sinalização gerados pelo CBT é menor que nos outros protocolos. A explicação está relacionada com a concepção do CBT, que não possui controle de fontes individuais, mas simplesmente o controle por grupo, desta forma reduzindo a quantidade dos pacotes de sinalização [RFC 2201].

6.2. CONTROLE DE ATRASOS MÉDIOS

O estabelecimento de uma fonte nos permite identificar o tempo gasto em milisegundos para que uma fonte se junte a um grupo *multicast* e comece a transmitir seus

pacotes, e o atraso fim-a-fim nos permite identificar o tempo médio necessário para que os *hosts* recebam um pacote *multicast* de uma determinada fonte.

Com esses duas medidas, apresentadas na Tabela 3 e na Figura 55, podemos identificar a rapidez da entrega *multicast* apresentada pelos protocolos investigados.

Protocolos	Atraso médio para Est. de Fonte	Atraso Médio Fim-a-Fim
MOSPF	3,89	2,76
PIM-SM	5,01	2,96
CBT	4,89	4,02

Tabela 3. Atraso médio para estabelecimento de uma fonte e atraso médio fim-a-fim

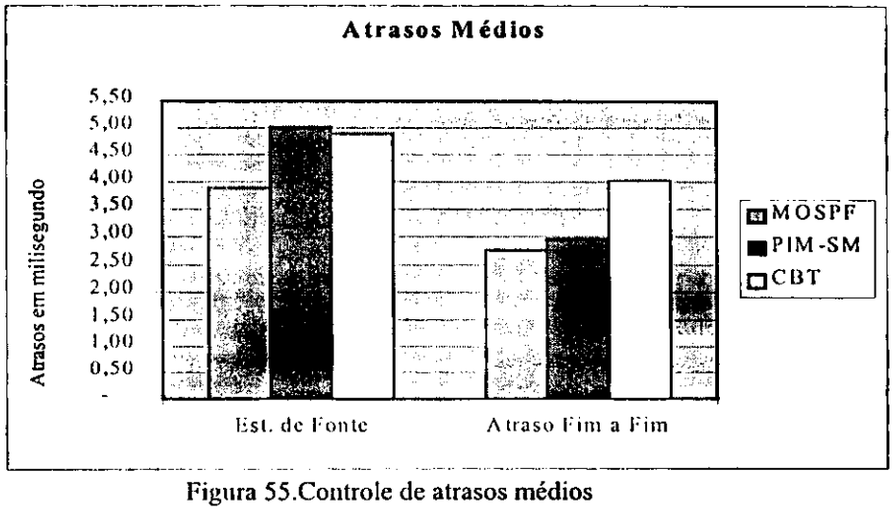


Figura 55.Controle de atrasos médios

Devido às trocas de caminho para a entrega de pacotes, primeiramente pela árvore RP e após pela árvore SPT, o PIM-SM apresenta o maior atraso médio para o estabelecimento de uma fonte, considerando que o simulador SPM foi desenvolvido para somar os dois atrasos referente às trocas mencionadas.

Outro fator que pode aumentar o atraso médio no estabelecimento de uma fonte é a implementação de árvores de distribuição pelo RP, já que os *hosts* devem anunciar ao RP que desejam ser fontes, e esse por sua vez é que vai disseminar a informação aos demais roteadores, causando um atraso médio maior do que se a própria fonte disseminasse a informação, como acontece com o MOSPF. Isso é uma das justificativas do atraso maior no CBT do que no MOSPF.

Observações semelhantes foram levantadas por [FARR 97] e [CALV 94], pois concluem em seus trabalhos que protocolos de modo denso apresentam menor atraso para estabelecimento de fontes, em relação aos outros protocolos, em função da utilização da técnica de inundação (*flooding*).

O atraso médio fim a fim, apresentado no CBT, justifica-se devido à utilização do RP, pois não usa o menor caminho e consequentemente sofre um maior atraso. Os outros dois protocolos se assemelham muito, já que na maior parte do tempo trabalham com a árvore SPT [CALV 94].

Uma observação importante é que os atrasos mencionados estão diretamente relacionados ao congestionamento da rede, número de grupos *multicast*, velocidade dos enlaces, tamanho dos grupos e posição geográfica dos *hosts* pertencentes a um grupo (grupos densos ou esparsos) .

6.3. COMPORTAMENTO DAS TABELAS DE ROTEAMENTO

Estas medidas, apresentadas na Tabela 4 e na Figura 56, nos permitem identificar o tamanho máximo que as tabelas de roteamento atingem em cada protocolo, permitindo desta forma, que sejam analisados os gastos de memória para armazenamento, velocidade de processamento para recuperação das informações e performance do sistema operacional para o encaminhamento correto e, com o menor atraso possível para os pacotes.

Protocolos	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	Média
MOSPF	119	122	124	167	123	187	172	183	142	121	146,00
PIM-SM	121	129	130	163	126	190	174	182	146	128	148,90
CBT	16	14	14	18	13	19	19	19	14	14	16,00

Tabela 4. Tamanho das tabelas de roteamento

O protocolo CBT implementa um controle de tráfego *multicast* por grupo, desta forma o máximo de entradas em uma tabela de roteamento é o número máximo de grupos simultâneos, que em nossa simulação é 19 [RFC 2201].

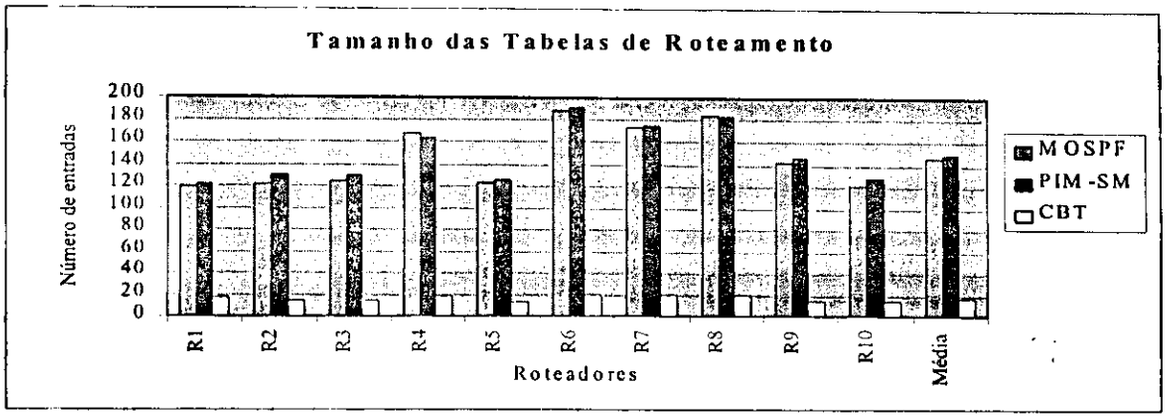


Figura 56. Tamanho das tabelas de roteamento

Os protocolos MOSPF e PIM-SM controlam o tráfego *multicast* por fonte, o que aumenta consideravelmente as entradas em uma tabela de roteamento [RFC 2117]. Sendo assim o tamanho máximo de uma tabela de roteamento para esses protocolos é a quantidade de grupos possíveis simultaneamente, multiplicado pelo número de roteadores que estão sendo considerados (levando em conta que todos os roteadores podem ter fontes de um determinado grupo ligadas diretamente a ele). No nosso caso temos 19 grupos e 10 roteadores, o que dá um total de 190 entradas no máximo para cada roteador.

Podemos observar que o roteador 6 é o que possui mais entradas em todos os protocolos, devido a sua posição geográfica no *backbone* e ao fato de ser escolhido como RP para o protocolo CBT e PIM-SM. Caso um outro roteador fosse escolhido como RP, teríamos variações nas entradas dos roteadores nos dois protocolos.

Resultados semelhantes foram encontrados por [FARR 97] e são utilizados como incentivo para o desenvolvimento do protocolo CBT de acordo com sua especificação em [RFC 2201].

O impacto ocasionado por tabelas grandes pode ser determinante para a baixa performance dos serviços *multicast*, alem de acarretar a necessidade do uso de equipamentos providos de muitos recursos computacionais e grande capacidade de armazenamento, tornando mais dispendiosa financeiramente a aquisição de novos equipamentos e inviabilizando a utilização de equipamentos antigos.

6.4. PACOTES POR ENLACE

Essas medidas, mostradas nas Tabelas 5, 6 e 7 e nas Figuras 58, 59 e 60, permitem a identificação da performance dos protocolos MOSPF, PIM-SM e CBT respectivamente, em relação a melhor distribuição do tráfego para atingir todos os destinos de um grupo *multicast*. Isso é importante para identificar quais os protocolos que geram maior saturação nos enlaces de uma rede.

Enlaces	Pac. Dados	Pac. Sinalização	*Overflow*	Total Pacotes
1:1-6	59.067	13.272	3	72.342
2:2-8	58.603	12.855	0	71.458
3:3-4	47.203	12.501	0	59.704
4:4-6	68.493	16.822	0	85.315
5:5-8	43.411	12.977	2.890	59.278
6:6-7	72.659	15.747	0	88.406
7:6-9	26.830	6.488	0	33.318
8:6-10	30.865	7.953	0	38.818
9:7-8	71.096	15.359	0	86.455
10:8-9	43.020	9.180	0	52.200
11:9-10	27.560	6.320	0	33.880

Tabela 5. Pacotes por enlace no MOSPF

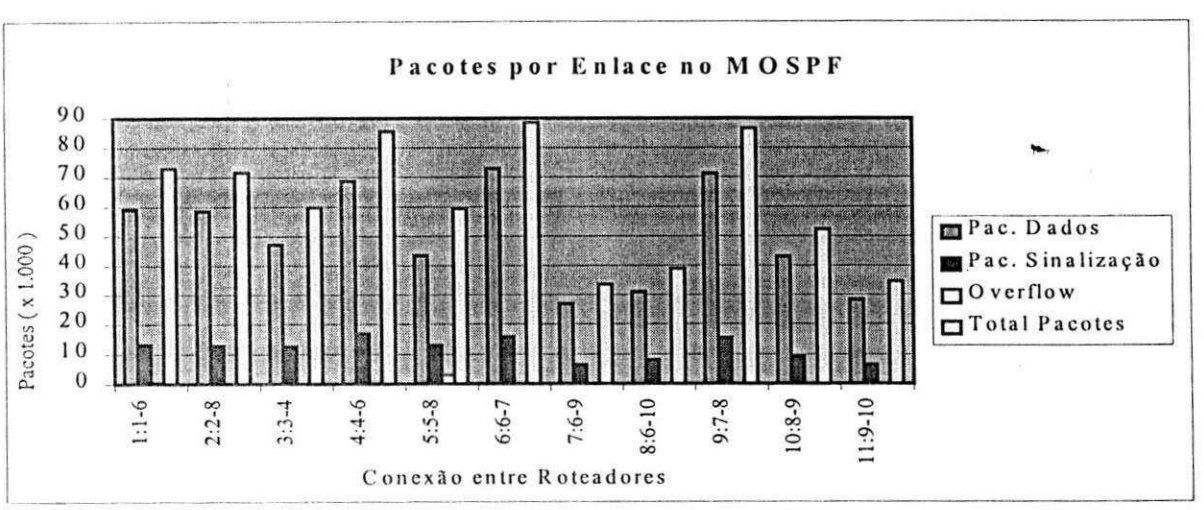


Figura 57. Pacotes por enlace no MOSPF

Enlaces	Pac. Dados	Pac. Sinalização	*Overflow*	Total Pacotes
1:1-6	58.743	13.310	0	72.053
2:2-8	57.275	12.859	0	70.134
3:3-4	47.087	12.473	0	59.560
4:4-6	68.601	16.750	0	85.351
5:5-8	43.273	12.831	2.849	58.953
6:6-7	74.212	16.076	0	90.288
7:6-9	26.057	6.683	0	32.740
8:6-10	32.643	8.012	0	40.655
9:7-8	71.387	15.412	0	86.799
10:8-9	42.407	8.883	0	51.290
11:9-10	29.175	6.077	0	35.252

Tabela 6. Pacotes por enlace no PIM-SM

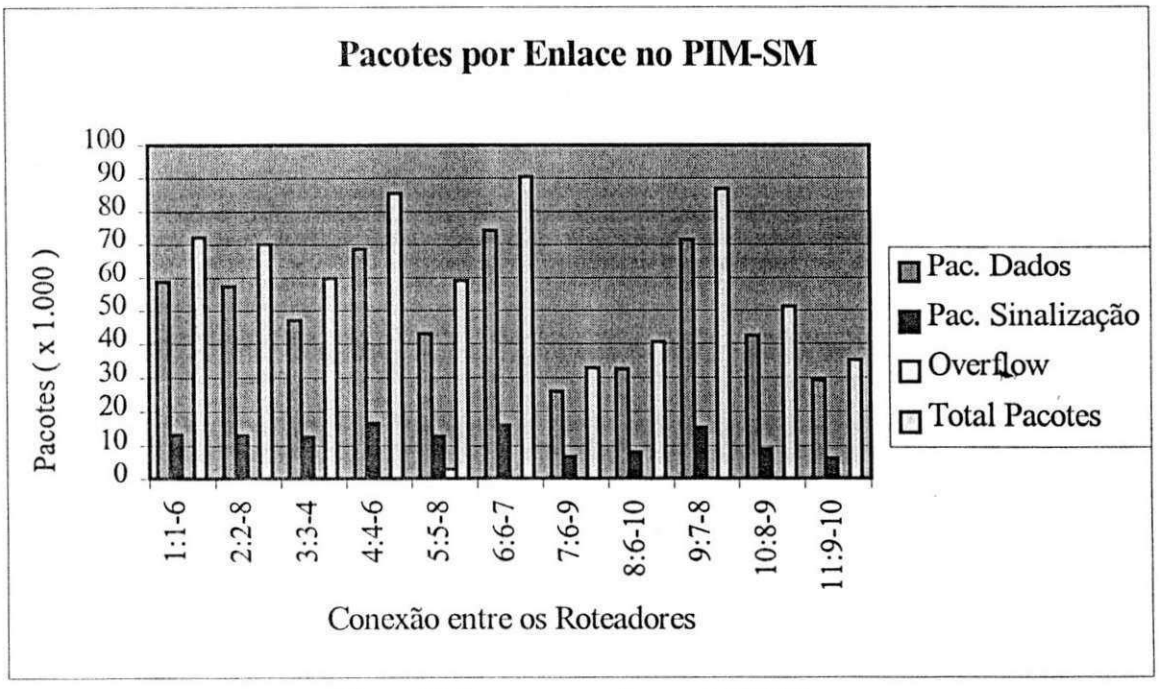


Figura 58. Pacotes por enlace no PIM-SM

Enlaces	Pac. Dados	Pac. Sinalização	*Overflow*	Total Pacotes
1:1-6	72.932	1.130	0	74.063
2:2-8	69.452	1.361	0	70.813
3:3-4	68.873	1.430	0	70.303
4:4-6	108.427	1.589	0	110.016
5:5-8	45.310	1.627	0	46.937
6:6-7	150.718	1.715	0	152.433
7:6-9	64.318	1.420	0	65.738
8:6-10	63.514	1.720	0	65.234
9:7-8	133.738	2.512	0	136.250
10:8-9	6.289	0	0	6.289
11:9-10	3.150	0	0	3.150

Tabela 7. Pacotes por enlace no CBT

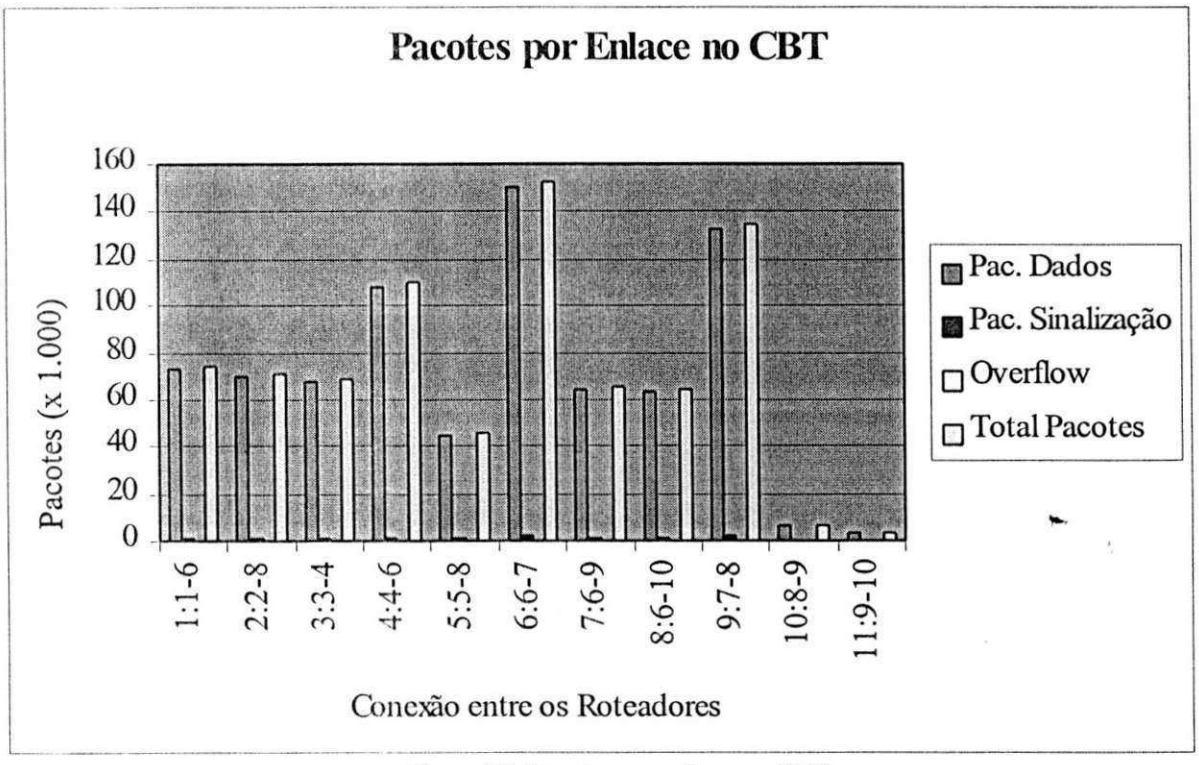


Figura 59. Pacotes por enlace no CBT

Os protocolos MOSPF e PIM-SM apresentam comportamentos semelhantes na distribuição do tráfego entre os enlaces, inclusive apresentam perdas de pacotes (*overflow*)

nos enlaces de baixa velocidade.

Por outro lado, o CBT apresenta uma concentração de tráfego muito grande em alguns enlaces, mas como os enlaces sobrecarregados são de alta velocidade, não ocorrem perdas de pacotes.

As observações de [FARR 97] e [CALV 94] também constatam que os protocolos baseados em árvores SPT distribuem o tráfego de forma mais homogênea entre os enlaces do que os protocolos baseados em árvores RP.

Estas observações nos levam a considerações sobre o projeto da rede em função da utilização ou não de tráfego *multicast* e, no caso de adoção deste serviço, a escolha do protocolo que melhor se adapte às restrições de velocidade dos enlaces disponíveis, de acordo com a forma de montagem das árvores *multicast* (árvore SPT, árvore RP ou forma híbrida).

O enlace ligando Rio de Janeiro a São Paulo (10:8-9) e o enlace ligando São Paulo a Porto Alegre (11:9-10) são pouco utilizados no CBT, pois não fazem parte da árvore de entrega *multicast*, montada pelo RP, desta forma, não são utilizados para a transferência de pacotes *multicast* e de sinalização, limitando-se apenas à transferência *unicast*.

Nos três protocolos, os enlaces mais utilizados foram: Recife a Brasília (4:4-6), Brasília a Belo Horizonte (6:6-7) e Belo Horizonte ao Rio de Janeiro (9:7-8).

Na Tabela 8 e na Figura 60, mostramos a ocupação do enlace entre Brasília e Belo Horizonte para salientarmos a sobrecarga que um enlace pode sofrer por protocolos que utilizam a concepção de RP permanente. Neste caso, o protocolo CBT apresenta aproximadamente o dobro da utilização dos outros dois protocolos.

Protocolos	Pac. Dados	Pac. Sinalização	Total Pacotes
MOSPF	72.659	15.747	88.406
PIM-SM	74.212	16.076	90.288
CBT	150.718	1.715	152.433

Tabela 8. Pacotes no enlace 6

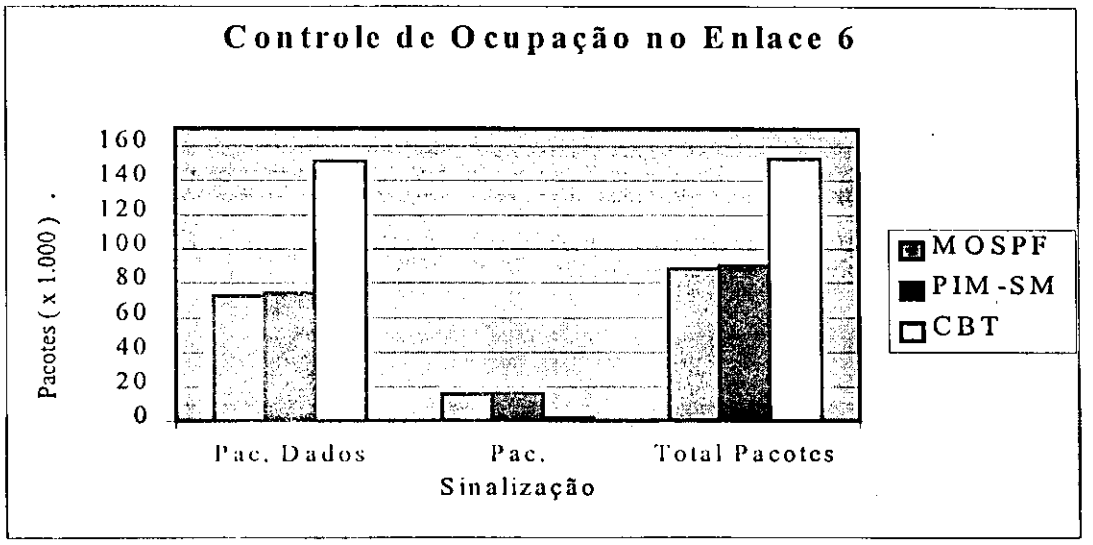


Figura 60. Pacotes no enlace entre Brasília e Belo Horizonte

Nesta seção, mostramos a diferença de sobrecarga sofrida pelos enlaces de acordo com cada protocolo, o que nos permite identificar a necessidade de estudos bem elaborados na hora de definirmos os RPs dos protocolos PIM-SM e CBT, já que deve ser levado em consideração a capacidade dos enlaces para a melhor distribuição das informações *multicast* [ZEGU 95].

CAPÍTULO 7

CONCLUSÕES E SUGESTÕES PARA FUTUROS TRABALHOS

Com o desenvolvimento das tecnologias de rede e sua adoção na concepção de novos serviços e produtos, a cada dia, convivemos mais freqüentemente com recursos computacionais interligados em rede, tornando indispensável a preocupação com a qualidade de serviço oferecida e a necessidade de novos protocolos que minimizem o desperdício de recursos numa rede.

De acordo com essa preocupação iminente, descrevemos nesse trabalho a técnica *multicast*, usada para a transferência de informações (pacotes) de uma origem a vários destinos, possibilitando o desenvolvimento de novos produtos e serviços, com baixo consumo de recursos.

Para experimentar o impacto do *multicast* em redes tipicamente de comutação de pacotes, como é o caso das que utilizam a família de protocolos TCP/IP, propomos e implementamos o simulador SPM que possibilita a avaliação da performance de modelos que representem os protocolos *multicast* de maneira eficiente.

Para definirmos um ambiente que trouxesse as características das redes TCP/IP, utilizamos a RNP como exemplo, com o intuito de analisar os novos protocolos e o

impacto causado por estes em alguns enlaces do cenário proposto, gerando desta forma, observações que possam vir a ser relevantes durante a implementação de *multicast* para permitir a passagem de tráfego multimídia.

A parametrização do SPM foi definida de forma a aproximar o ambiente de simulação do mundo real, sendo assim, o simulador apresenta a possibilidade de trabalhar com parâmetros de entrada, permitindo que estes sejam especificados de acordo com qualquer cenário proposto (visto no capítulo 4).

Como o SPM foi construído de forma modular, este permite que modelos de protocolos sejam acoplados a ele sem que necessite de grandes modificações, desta forma, encontra-se estruturado para a avaliação de outros protocolos que possam ser desenvolvidos.

A utilização de estruturas de dados adequadas, permitem que outras pessoas possam modificar o código fonte a fim de estender as funcionalidades desse simulador.

Os pontos de investigação são colocados em locais estratégicos do simulador, para permitir que, independente da implementação de novos protocolos, os resultados possam ser armazenados e apresentados sem a necessidade de modificações no núcleo do simulador.

Por ser um simulador de fim específico, a execução de uma simulação é executada rapidamente (em minutos, para nosso exemplos apresentados), ao contrário do que acontece na utilização de simuladores de fins gerais onde a obtenção dos resultados é muito demorada podendo levar até dias, de acordo com os parâmetros de entrada propostos na simulação.

Pela análise feita, após os resultados das simulação (ver capítulo 6), apresentamos abaixo as principais observações obtidas em relação aos protocolos simulados e ao cenário proposto neste trabalho:

- As árvores de distribuição *multicast*, montadas pelo menor caminho, são mais eficientes que as árvores RP, devido ao menor número de replicações que um pacote *multicast* sofre para atingir o seu destino;

- Os protocolos que controlam o tráfego *multicast*, de acordo com a fonte e grupo, precisam trocar mais informações (geram mais pacotes de sinalização) entre os roteadores

do que os protocolos que controlam o tráfego *multicast* de acordo com o grupo simplesmente. Entretanto, perdem em termos de flexibilidade no caso de determinados *hosts* necessitarem receber tráfego *multicast* de algumas fontes apenas, e não de todas as fontes que compõem o grupo;

- O atraso para estabelecimento de fontes aumenta caso haja um intermediário (RP) entre a fonte e os demais *hosts*, já que existe a necessidade da disseminação da informação pelo RP e não diretamente da nova fonte;

- No caso de protocolos que mudam a árvore de roteamento *multicast* após uma determinada carga de tráfego, o atraso para estabelecimento de fontes aumenta, pois é necessário estabelecer duas vezes o caminho até os membros;

- O atraso médio fim-a-fim está relacionado com o congestionamento dos enlaces e com o caminho percorrido pelos pacotes, desta forma, protocolos que implementam RP apresentam maior atraso médio fim-a-fim devido à concentração de tráfego em determinados enlaces e, na maioria dos casos, por não utilizarem o menor caminho para a entrega dos pacotes da origem até o destino;

- A memória que um roteador deve possuir para armazenar as informações de roteamento está relacionada com o número de entradas que uma tabela possa vir a ter. Dessa forma, os protocolos que possuem controle de tráfego *multicast* por grupo e fonte necessitam de muito mais memória que os protocolos que controlam o tráfego por grupo;

- Outra consideração importante é a velocidade de processamento que um roteador deve possuir no caso de pesquisas em tabelas que são muito grandes. A performance de roteadores implementados com o MOSPF e PIM-SM deve ser alta, para não haver problemas de processamento e possível perda de pacotes que possuem restrição de tempo para chegarem a seus destinos;

- Árvores SPT aproveitam melhor os enlaces disponíveis em uma topologia, por outro lado, *Rendezvous Point* pode forçar o tráfego a concentrar-se em enlaces de maior velocidade;

• Protocolos que utilizam *Rendezvous Point* apresentam sobrecarga de certos enlaces, o que pode ser perigoso em caso de algum tipo de falha no enlace ou roteador. Embora existam RPs alternativos, muitos pacotes podem ser perdidos em caso de falha, mas por outro lado, como temos uma centralização das atividades, a administração e manutenção pode ser mais rápida e fácil.

A seguir, na Tabela 9, apresentamos um quadro que compara os três protocolos abordados:

Pontos Investigados	MOSPF	PIM-SM	CBT
Replicação de pacotes	Baixo	Médio	Alto
Tráfego de sinalização	Alto	Alto	Baixo
Atraso médio para Estabelecimento de Fonte	Baixo	Alto	Médio
Atraso médio Fim a Fim	Baixo	Baixo	Alto
Utilização de recursos do roteador	Alto	Alto	Baixo
Concentração de Tráfego	Baixo	Baixo	Alto
Complexidade de implementação	Médio	Alto	Baixo

Tabela 9. Comparação entre os protocolos abordados

Com os resultados obtidos, mostramos que o simulador SPM é adequado para simular redes TCP/IP implementadas com tráfego *multicast* e que este poderá contribuir para avaliação de desempenho de redes rodando *multicast*, além de ser uma ferramenta útil na hora da escolha de um dos protocolos existentes para ser implementado em redes reais.

Nos próximos anos, acreditamos que inúmeras redes no mundo serão implementadas com recursos *multicast*, inicialmente nas *Intranets* e após na *Internet* [HURW 97], assim novos trabalhos devem ser desenvolvidos para dar continuidade às pesquisas hoje em andamento, neste sentido, listamos a seguir algumas sugestões e planos que estamos traçando para desenvolvimento futuro:

• Estender o SPM para que possa simular o modelo dos protocolos DVMRP e PIM-DM, a fim de permitir que este possa avaliar todos os protocolos existentes;

• Estender o SPM para que permita a visualização gráfica dos resultados, facilitando assim as avaliações das simulações;

- Implementar no SPM um gerenciador de memória ou paginação em disco rígido, possibilitando a simulação de grandes topologias que envolvam centenas de nodos e grupos;

- Utilizar o SPM para avaliar outros cenários, com variação dos parâmetros de entrada, permitindo uma análise mais consistente dos protocolos *multicast*;

- Configurar os protocolos *multicast* em roteadores atualmente no mercado, permitindo a construção de documentos que possam facilitar a implementação desse serviços em redes reais, com a intenção de colaborar com a comunidade TCP/IP.

Este documento não tem a finalidade de esgotar um assunto tão amplo como é a técnica *multicast*, entretanto esperamos que sirva como referência bibliográfica para futuros trabalhos nessa área, além de possibilitar a utilização e ampliação do simulador SPM, em vista da grande utilização *multicast* que ocorrerá nos próximos anos.

REFERÊNCIAS BIBLIOGRÁFICAS

[ALTA 95] **Alta Group.** *"BONeS DESIGNER - An Introduction"*. For Use with DESIGNER Software Release 3.0 or Higher.

[BRAND 96] **Brandão, J. E. M. S.** *"Gerência de Recursos em Redes Compartilhadas com Integração de Serviços - Uma Proposta e Implementação"*. Dissertação de Mestrado, Universidade Federal da Paraíba, Agosto de 1996.

[CAIN 97] **Cain, B. & Deering S. & Thyagarajan, A.** *"Internet Group Management Protocol, Version 3"*. IETF Draft draft-ietf-idmr-igmp-v3-00.txt, Nov 1997.

[CALV 94] **Calvert, K. & Madhavan R. & Zegura, E.** *"A Comparison of Two Practical Multicast Routing Schemes"*. College of Computing, Georgia Intitute de Technology, Atlanta, February 1994.

[CARV 96] **Carvalho, T. & Nunes, A. & Rech, A. & Medeiros, B. & Perpingnam, D. & Specialski, E. & Mota, F. & Reale, G. & Brandão, H.** *"Arquitetura de Redes de Computadores OSI e TCP/IP"*. Editora Makron Books, Rio de Janeiro, 1994.

[CASNE 96] **Casner, S.** *"Frequently Asked Questions (FAQ) on the Multicast Backbone (MBONE)"*. URL = http://www.research.att.com/mbone-faq.html, ou ftp://ftp.isi.edu/ mbone/faq.txt.

[COMER 91] **Comer, D. E.,** *"Internetworking with TCP/IP - Volume I; Principles, Protocols, And Architecture - Second Edition"*. Prentice Hall International

Editions, 1991.

[ERIK 97] **Eriksson, S.** *"MBONE: The Multicast Backbone"*. URL = http://www.mang.canterbury.ac.nz/%7Ebusa057/mbone/art1.html, 1997.

[ESTR 97] **Estrin, D. & Helmy, A. & Thaler, D.** *"PIM Multicast Border Router (PMBR) specification for Connecting PIM-SM domains to a DVMRP Backbone"*. IETF Draft, Draft-ietf-mbone-pmbr-spec-00.txt, Feb 1997.

[FARR 97] **Farrey-Goudreau, E. & Billhartz T. & Cain, B. & Fieg, D. & Batsell, S.** *"Performance and Resource Cost Comparisons for the CBT and PIM Multicast Routing Protocols"*. This work was sponsored by the Defense Advenced Research Projects Agency (DARPA) through contract number N00014-93-C-2186 with the Naval Research Laboratory.

[FENNE 97] **Fenner. W.** *"Internet Group Management Protocol, Version 2"*. IETF Draft Draft-ietf-idmr-igmp-v2-08.txt, Nov 1997.

[FIRO 95] **Firoiu. V. & Towsley, D.** *"Call Admission and Resource Reservation for Multicast Sessions"*. Computer Science Department, University of Massachusetts, Technical Report TR95-17, September 1995.

[HARR 95] **Harrison, T.** *"Performance Evaluation of Routing Algorithms for Multicast Traffic"*. A Thesis Submitted to the College of Graduate Studies and Research in Partial Fulfillment of the Requirements for the Degree of Master of Science in the Department of Computer Science, University of Saskatchewan, August 1995.

[HAWK 97] **Hawkinson, J.** *"Multicast Pruning a Necessity"*. IETF Draft, Draft-ietf-mboned-pruning-02.txt, July 1997.

[HELMY 97] **Helmy, A. & Deering, S. & Farinacci, D. & Jacobson, V. & Estrin, D. & WEI, L.** *"Protocol Independent Multicast version 2, Dense Mode Specification"*. IETF Draft, Draft-ietf-idmr-pim-dm-05.txt, May 1997.

[HURW 97] **Hurwicz, M.** *"Multicast para a Massa - O padrão IP multicast está pronto, mas a infra-estrutura não. Ainda."*. Byte, Julho 1997.

[IM 95] **Im, Y. & Lee, Y. & Wi, S. & Lee, K. & Choi, Y. & Kim, C.** *"Multicast Routing Algorithms in High Speed Networks"*. Dept. of Computer Engineering, Seoul National University, December 1995.

[JOHNS 98] **Johnson, V. & Johnson, M.** *"Higher Level Protocols Used With IP Multicast"*. URL = http://www.ipmulticast.com/community/whitepapers/highprot..html.

[LEE 97] **Lee, Y. & Im, Y. & Lee, K. & Choi, Y.** *"A Bandwidth and Delay Constrained Minimum Cost Multicast Routing Algorithm"*. Dept. of Computer Enginnering, Seoul National University, Corea 1997.

[MALK 97] **Malkin, G.** *"RIP Version 2"*. Draft-ietf-ripv2-protocol-v2-02.txt, March 1997.

[MOURA 86] **Moura, J. A. B. & Sauvé, J. P. & Giozza, W. F. & Araújo, J. F. M.** *"Redes Locais de Computadores - Protocolos de Alto Nível e Avaliação de Desempenho"*. McGraw-Hill - Ltda, 1986.

[NORO 94A] **Noronha, C. A.** *"Routing of Video/Audio Streams In Packet-Switched Networks"*.Technical Report No. CSL-TR-94-653, Departments of Electrical Engeneering and Computer Science, Stanford University, December 1994.

[NORO 94B] **Noronha, C. A. & Tobagi, A. T.** *"Evaluation of Multicast Routing Algorithms for Multimedia Streams"*. IEEE ITS, August 1994.

[PUSAT 96] **Pusateri, T.** *"Distance Vector Multicast Routing Protocol - Version 3"*. IETF Daft Draft-ietf-idmr-dvmrp-v3-05.txt , octuber 1997.

[RFC 0768] **Postel, J.** *"User Datagram Protocol"*. RFC 0768, 1980.

[RFC 0792] **Postel, J.** *"Internet Control Message Protocol"*. RFC 0792, 1981.

[RFC 1075] **WAITZMAN, D. & PARTIRIDGE, C. & Deering, S.** *"Distance Vector Multicast Routing Protocol"*. RFC 1075, november 1988.

[RFC 1112] **Deering, S.** *"Host Extension for IP Multicasting"*. RFC 1112, 1989.

[RFC 1301] **Freier, A. & Marzullo, K.** *"Multicast Transport Protocol"*. RFC 1301, February 1992.

[RFC 1371] **Gross, P.** *"Choosing a Common IGP for the IP Internet"*. RFC 1371, October 1992.

[RFC1584] **Moy, J.** *"Multicast Extensions to OSPF"*. RFC 1584, March 1994.

[RFC1585] **Moy, J.** *"MOSPF: Analysis and Experience"*. RFC 1585, March 1994.

[RFC1889] **Schulzrinne, H. & Casner, S. & Frederick, R. & Jacobson, V.** *"RTP: A Transport Protocol for Real-Time Applications"*. RFC 1889, January 1996.

[RFC1890] **Schulzrinne, H.** *"RTP Profile for Audio and Video Conferences with Minimal Control"*. RFC 1890, January 1996.

[RFC 2117] **Deering, S. & Estrin, D. & Farinacci, D. & Handley, M. & Helmy, A. & Jacobson, V. & Liu, C. & Sharma, P. & Thaler, D. & Wei, L.** *"Protocol Independent Multicast-Sparse Mode (PIM-SM) : Protocol Specification"*. RFC 2117, June 1997.

[RFC 2189] **Ballardie, A.** *"Core Based Trees (CBT version 2) Multicast Routing"*. RFC 2189, September 1997.

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

[RFC 2210] **Wroclawski, J.** *"The Use of RSVP with IETF Integrated Services"*. RFC 2210, September 1997.

[RNP 98] **Ministério da Ciência e Tecnologia - Centro de Informações.** *"Descrição*

do Projeto". http:// www.rnp.br/1.3.desc.html, 1997.

[SCHIL 91] **Schildt, H.** *"C Completo e Total"*. Editora McGrawn-Hill, São Paulo, 1991.

[SEMER 97] **Semeria, C. & Maufer, T.** "*Introduction to IP Multicast Routing*". Internet-Draft, Draft-ietf-mbone-intro-multicast-03.txt, july 1997.

[SOUTO 94] **SOUTO, F. A. C.**, *"SAVAD - Sistema de Avaliação de Desempenho de Modelos de Redes de Filas"*. Dissertação de Mestrado, Campina Grande, novembro de 1994.

[STANT 96] **Stanton, M. A. & Barra, L. F. S. & Bastos, C. A. M.** *"Integração de Serviços na Internet"*. 14º Simpósio Brasileiro de Redes de Computadores, Maio de 1996.

[TANEN 95] **Tenenbaum, A. & Langsam, Y. & Augenstein, M.** *"Estrutura de Dados Usando C"*. MAKRON Books, São Paulo, 1995.

[THYA 96] **Thyagarajan, A. & Deering, S.** *"Hierarchical Distance-Vector Multicast Routing for the MBone"*. Department of Electrical Engineering, University of Delaware, Delaware 1996.

[ZEGU 95] **Zegura, E. & Calvert, K. & Donahoo, M.** *"Core Selection Methods for Multicast Routing"*. College of Computing, Georgia Institute of Technology, Atlanta 1995.