HBH: Um Protocolo de Roteamento para a Implantação Progressiva do Serviço Multicast

Luís Henrique M. K. Costa[1,2] Serge Fdida[1] Otto Carlos M. B. Duarte[2]

`Luis.Costa@lip6.fr` `Serge.Fdida@lip6.fr` `otto@gta.ufrj.br`

[1] *LIP6 - Université Pierre et Marie Curie*

4, place Jussieu - 75252 - Paris Cedex 05 - France

[2] *GTA/COPPE/EE - Universidade Federal do Rio de Janeiro*

C.P. 68504 - 21945-970 - Rio de Janeiro, RJ - Brasil *

Resumo

Apesar de uma década de pesquisa desde a proposta original da arquitetura IP Multicast, sua implantação na Internet está apenas começando. Como conseqüência, a Internet tende a conviver com redes unicast e multicast. Desta forma torna-se importante o desenvolvimento de protocolos que permitam a implantação progressiva do serviço multicast através do suporte de nuvens unicast. Este artigo propõe HBH (*Hop-By-Hop multicast routing protocol*). HBH adota a abstração de canal específico à fonte para simplificar a alocação de endereços e implementa a distribuição de dados através de unicast recursivo, que permite o suporte transparente de roteadores puramente unicast. Além disso, HBH é original porque seu mecanismo de construção de árvores leva em conta as assimetrias do roteamento unicast. Como a maioria dos protocolos multicast se apóia na infra-estrutura unicast, essas assimetrias podem afetar a qualidade das árvores multicast. As simulações realizadas mostram que HBH possui desempenho superior a outros protocolos em termos do atraso experimentado pelos receptores e da banda passante consumida pelas árvores multicast.

Abstract

IP multicast has been addressed since more than a decade but very little has been achieved as far as deployment is concerned. As a consequence, the Internet is likely to be organized with both unicast and multicast enabled networks. Therefore, it is of utmost importance to design protocols that allow the progressive deployment of the multicast service by supporting unicast clouds. This paper proposes HBH (Hop-By-Hop multicast routing protocol). HBH adopts the source-specific channel abstraction to simplify address allocation and implements data distribution using recursive unicast trees, which allow the transparent support of unicast-only routers. Additionally, HBH is original because its tree construction algorithm takes into account the unicast routing asymmetries. As most multicast routing protocols rely on the unicast infrastructure, these asymmetries impact the quality of the multicast trees. Our simulations show that HBH outperforms other routing protocols in terms of the delay experienced by the receivers and the bandwidth consumption of the multicast trees.

Palavras-chave: Internet, IP Multicast, protocolos de roteamento, comunicação de grupo, qualidade de serviço.

*Este trabalho foi patrocinado por FUJB, CNPq, CAPES/COFECUB e IST Project GCAP N^0 1999-10504.

1 Introdução

Apesar de uma década de pesquisa desde a proposta original da arquitetura IP Multicast [1], sua implantação na Internet está apenas começando. Vários fatores frearam seu desenvolvimento. A arquitetura é composta de um modelo de serviço que define um grupo como uma conversação aberta entre M fontes e N receptores, um esquema de endereçamento baseado em endereços IP classe D e protocolos de roteamento. Qualquer estação pode enviar dados assim como conectar-se ao grupo e ter acesso à informação. O modelo IP Unicast também é completamente aberto, porém no multicast os estragos causados por uma fonte intrusa são multiplicados pelo tamanho do grupo.

Um grupo multicast é identificado por um endereço IP classe D que não é diretamente relacionado a nenhuma informação geográfica como no roteamento hierárquico unicast. A alocação de endereços é portanto complicada assim como sua agregação nas tabelas de roteamento. Atualmente não existe solução escalável de roteamento inter-domínio.

Apesar deste cenário, os provedores de serviço Internet (ISPs) têm interesse no serviço multicast como resposta à crescente demanda por banda passante e novas aplicações de distribuição de conteúdo. Como conseqüência, a Internet tende a possuir redes puramente unicast convivendo com redes multicast. Torna-se muito importante, portanto, o desenvolvimento de protocolos capazes de permitir a implantação progressiva do serviço multicast através de "nuvens" unicast.

Várias soluções que simplificam o serviço multicast através da redução do modelo de distribuição foram propostas [2]. EXPRESS [3] restringe a conversação multicast a *1 para N* (introduzindo a abstração de *canal*), o que simplifica a alocação de endereços e a distribuição de dados. Além disso, a maioria das aplicações atuais possuem apenas uma fonte ou um pequeno número de fontes. No entanto, este serviço específico à fonte não soluciona o problema de desenvolvimento *progressivo* do serviço multicast. Atualmente, a única solução para o multicast atravessar uma rede unicast é a utilização de túneis.

A capacidade de suportar roteadores unicast de forma transparente é a principal motivação de HBH (*Hop-By-Hop multicast routing protocol*) que este artigo propõe. HBH implementa a distribuição de dados através de árvores unicast recursivas, proposta original de REUNITE [4]. REUNITE não usa endereços IP classe D para identificação do grupo, abandonando completamente o esquema de endereçamento IP Multicast.

HBH usa a infra-estrutura unicast diminuir o tamanho das tabelas de de roteamento da mesma forma que REUNITE, mas utilizando o modelo de canal de EXPRESS. Desta forma a compatibilidade com IP Multicast é garantida. Além disso, HBH constrói árvores de tipo SPT (*Shortest-Path Trees*), onde cada receptor recebe dados da fonte através do caminho mais curto fonte-receptor, ao contrário da maioria dos outros protocolos que constróem árvores SPT *reversas* [5, 6, 7, 8]. Desta forma, HBH tem o potencial de construir árvores de melhor qualidade na presença de redes assimétricas e é melhor adaptado a uma eventual implementação de roteamento baseado em Qualidade de Serviço (QoS). Além disso, seu mecanismo de construção de árvores proporciona uma maior estabilidade face à dinâmica do grupo e menor consumo de banda passante comparado a REUNITE.

O artigo é organizado da seguinte forma: a Seção 2 apresenta a pesquisa relacionada, motivações e idéias de base de HBH, a Seção 3 descreve o funcionamento de HBH, a Seção 4 apresenta uma comparação de desempenho entre HBH e outros protocolos multicast e a Seção 5 apresenta as conclusões deste trabalho.

2 Os Princípios de Base de Hop-By-Hop Multicast

Esta seção faz uma breve apresentação dos protocolos EXPRESS e REUNITE para em seguida apresentar os princípios de base de HBH assim como os problemas que o roteamento unicast assimétrico pode causar, e que motivaram a concepção de HBH.

2.1 Trabalhos relacionados

O protocolo EXPRESS [3] propõe uma solução simples para o problema de alocação de endereços através da introdução do modelo de canal que reduz o modelo de distribuição de *M para N* para *1 para N*. Um canal é identificado pelo par $<S, G>$ onde S é o endereço unicast da fonte e G é um endereço multicast (IP classe D). A concatenação destes dois endereços resolve o problema de alocação de endereços de grupo porque o endereço unicast é por definição único. O modelo de canal também simplifica a implantação de funções de gestão de grupo como o controle do acesso de fontes ao grupo, embora a implementação em curso do protocolo (PIM-SSM [9]) não trate estes aspectos.

O protocolo REUNITE (REcursive UNIcast TrEes) [4] implementa a distribuição multicast baseada na infra-estrutura unicast. A principal motivação de REUNITE é que a forma típica das árvores multicast é “alongada” – a maioria dos roteadores simplesmente recebe os pacotes e os reenvia em uma única interface de saída. Em outras palavras, a minoria dos roteadores são nós de ramificação da árvore. Entretanto, os protocolos atuais mantém entradas nas tabelas de reenvio de pacotes para todos os grupos que atravessam o roteador. A idéia é então de separar a informação de roteamento em duas tabelas: uma tabela de controle multicast (MCT - *Multicast Control Table*) e uma tabela de reenvio multicast (MFT - *Multicast Forwarding Table*), a primeira localizada no plano de controle e a segunda no plano de dados do roteador. Roteadores de simples reenvio (uma única saída) para um determinado grupo mantém informação para este grupo apenas em sua MCT, enquanto nós de ramificação possuem entradas em sua MFT que são utilizadas para recursivamente criar cópias dos pacotes de forma a servir todos os membros do grupo.

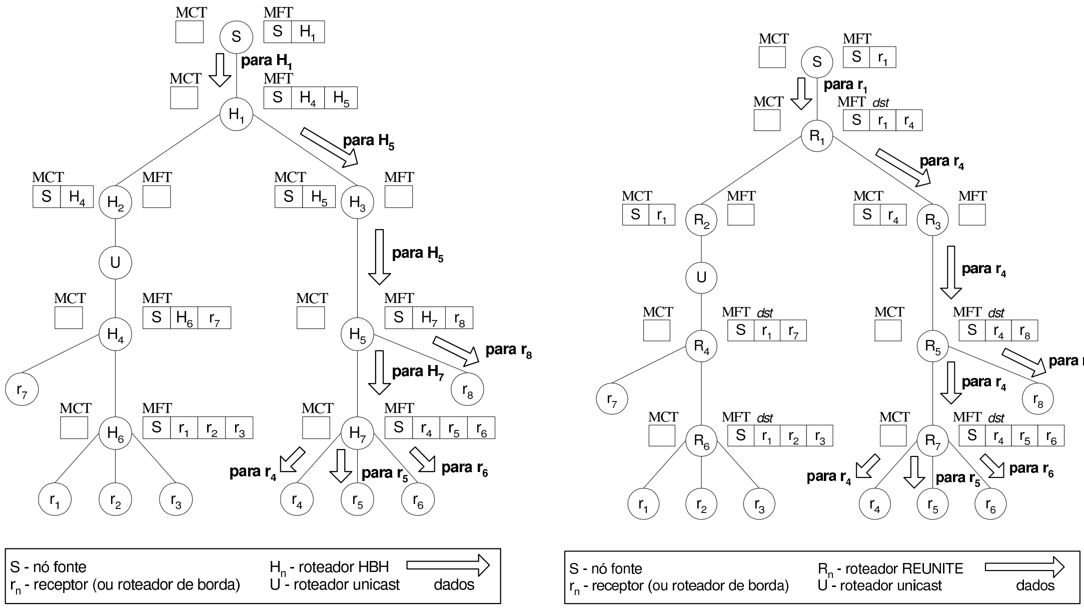
REUNITE identifica a conversação por um par $<S, P>$, onde S é o endereço unicast da fonte e P é um número de porta alocado pela fonte. Não são usados endereços IP classe D. As tabelas de roteamento são preenchidas de acordo com a chegada de receptores ao grupo. REUNITE utiliza dois tipos de mensagem, *join* e *tree*, para manter o *soft-state* da árvore. As mensagens *join* são periodicamente enviadas pelos receptores na direção da fonte, enquanto as mensagens *tree* são periodicamente produzidas pela fonte e enviadas em multicast de forma a atualizar o *soft-state* da árvore de distribuição. Apenas os nós de ramificação para o grupo $<S_1, P_1>$ mantém entradas $<S_1, P_1>$ em suas MFTs. A tabela de controle, MCT, não é utilizada para o reenvio de pacotes. Roteadores de simples reenvio na árvore $<S_1, P_1>$ possuem entradas na MCT mas nenhuma entrada na MFT.

2.2 A distribuição multicast através de unicast recursivo

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

A Figura 1(a) mostra como o unicast recursivo é utilizado pelo protocolo HBH. A fonte S produz dados que são endereçados a H_1. H_1 cria duas cópias de cada pacote que são enviadas a H_4 e H_5 (os próximos nós de ramificação). H_3 simplesmente reenvia os pacotes em unicast. H_5 recebe os dados e envia cópias para H_7 e r_8. Finalmente, H_7 cria uma cópia do pacote para r_4, r_5 e r_6. A distribuição dos dados é simétrica do outro lado da árvore.

A Figura 1(b) ilustra a distribuição de dados no protocolo REUNITE. A fonte envia os dados endereçados ao primeiro membro que se conectou ao grupo. Em um nó de ramificação, R_B, os pacotes recebidos possuem como endereço de destino o endereço do primeiro receptor, r_i, que se conectou ao grupo na sub-árvore abaixo de R_B. r_i é armazenado em uma entrada especial da MFT, MFT$<S>.dst$. R_B cria uma cópia dos dados para cada receptor presente em sua MFT (o endereço de destino de cada cópia é igual ao endereço unicast do receptor). A cópia original do pacote é reenviada a r_i. Neste exemplo, S produz pacotes de dados endereçados a r_1 (estes pacotes chegam a r_1 inalterados). R_1 cria uma cópia do pacote com endereço de destino r_4. R_3 simplesmente reenvia os pacotes sem precisar consultar sua MFT. R_5 cria uma cópia do pacote para r_8 e finalmente R_7 cria cópias para r_5 e r_6.



(a) Árvore HBH.

(b) Árvore REUNITE.

Figura 1: Distribuição de dados através de unicast recursivo.

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

2.3 Os riscos do roteamento assimétrico

Roteamento assimétrico significa que o caminho unicast entre A e B pode ser diferente do caminho entre B e A. Este fenômeno pode ocorrer devido a diversas razões. O caso mais simples é o de enlaces assimétricos ou unidirecionais, como por exemplo linhas ADSL ou enlaces de satélite. No entanto existem outras fontes de rotas assimétricas, como roteadores mal configurados ou rotas configuradas assimétricas intencionalmente. Esta configuração (conhecida como "hot-potato routing") é uma forma de minimizar a utilização de sua rede como trânsito entre redes de terceiros.

O roteamento unicast assimétrico afeta o roteamento multicast porque a maioria dos protocolos de roteamento multicast constrói árvores de tipo SPT reversa (*reverse Shortest-Path Tree*)[5, 6, 7]. Neste caso os pacotes de dados enviados pela fonte ao receptor seguem a rota unicast utilizada para ir do receptor à fonte. Se as rotas de ida e volta possuem características diferentes, por ex. de atraso, o uso da árvore SPT reversa pode ser problemático para a implementação de QoS. A capacidade de construir árvores SPT é portanto vantajosa para o protocolo de roteamento.

REUNITE é uma proposta original justamente por (potencialmente) construir árvores SPT. (MOSPF - *Multicast Open Shortest Path First* [10] é o único protocolo Internet a construir SPTs.) Isto é possível em REUNITE porque as mensagens *tree* que trafegam da fonte para os receptores instalam as entradas nas tabelas de reenvio de pacotes e não as mensagens *join* que seguem a direção inversa. Entretanto, REUNITE pode falhar na construção da árvore SPT devido a assimetrias no roteamento unicast. REUNITE possui outro inconveniente: a rota utilizada por um receptor pode mudar após a saída de outro membro do grupo. Isto dificulta a implementação de mecanismos de QoS.

A Figura 2 ilustra o mecanismo de construção da árvore REUNITE com um exemplo onde este falha na construção da SPT. Considere as rotas unicast: $r_1{>}R_2{>}R_1{>}S$; $S{>}R_1{>}R_3{>}r_1$; $r_2{>}R_3{>}R_1{>}S$; $S{>}R_4{>}r_2$. Suponha os seguintes eventos: r_1 se conecta a $<S,P>$, r_2 se conecta a $<S,P>$ e r_1 deixa o grupo.

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

Nesta configuração, r_1 recebe os dados de S através do caminho mais curto, mas não r_2. Uma vez que as rotas unicast entre S e r_2 são assimétricas e como R_3 intercepta o $join(S,r_2)$, os dados seguem o caminho $S{>}R_1{>}R_3{>}r_2$, o mesmo caminho utilizado pelas mensagens *tree* para irem da fonte S a r_2 (Figura 2(a)).

Os estados mantidos na MCT e MFT são *soft-state*. Os receptores periodicamente enviam mensagens $join(S,r_i)$ e a fonte periodicamente produz uma mensagem $tree(S,r_i)$

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

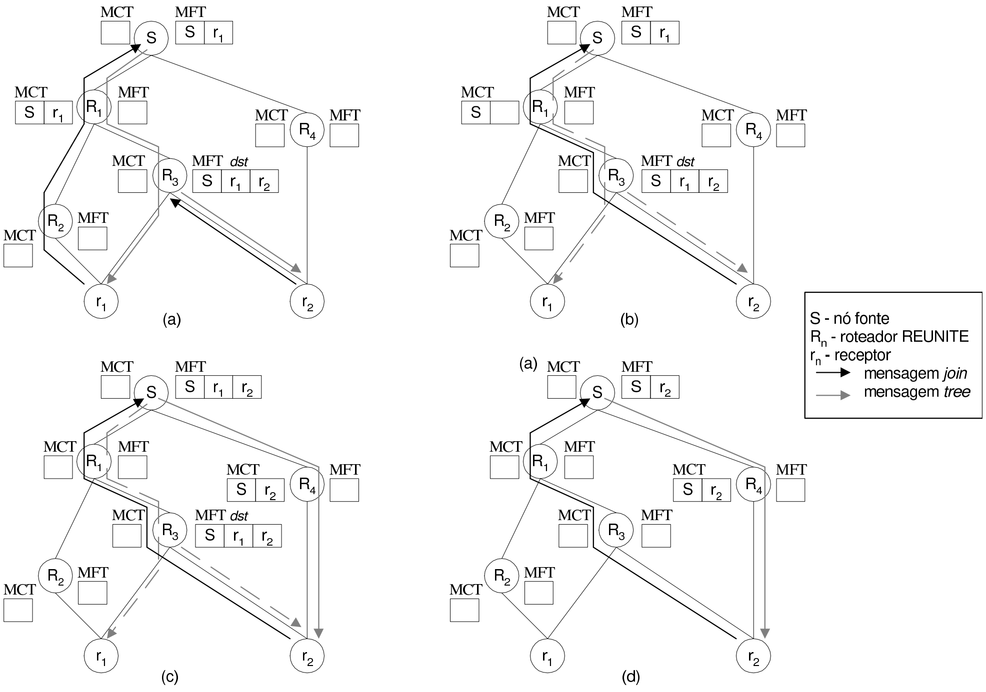


Figura 2: O mecanismo de construção da árvore REUNITE.

em multicast. Para se desconectar do canal o receptor deve simplesmente parar o envio de mensagens *join*. Quando a árvore está estabilizada, os $tree(S, r_i)$ atualizam o *soft-state* de entradas r_i nas MCTs dos roteadores assim como as entradas MFT$<S>.dst = r_i$. Os $join(S, r_j)$ atualizam a entrada r_j na MFT do nó onde r_j se conectou a $<S>$ (na Figura 2, os $join(S, r_1)$ atualizam r_1 na MFT de S e os $join(S, r_2)$ atualizam r_2 na MFT de R_3).

Suponha agora que r_1 deixa o grupo, parando de emitir $join(S, r_1)$. Como a entrada r_1 na MFT de S deixa de ser atualizada, após a expiração do temporizador *t1* a entrada r_1 se torna *stale*. Um segundo temporizador, *t2*, é criado e vai destruir a entrada r_1 caso esta não seja mais atualizada. Uma vez que r_1 está *stale*, S envia mensagens $tree(S, r_1)$ *marcadas* (Figura 2(b)). Mensagens $tree(S, r_1)$ marcadas significam que o fluxo de dados endereçados a r_1 vai cessar em breve, logo a parte da árvore contendo r_1 nas tabelas de roteamento deve ser reconfigurada. As MFT nos nós de ramificação que possuem MFT$<S>.dst = r_1$ tornam-se *stale* após a recepção das mensagens *tree* marcadas. Em nós de simples reenvio, a recepção de $tree(S, r_1)$ marcadas causa a destruição de entradas r_1 da MCT. Conseqüentemente, os $join(S, r_2)$ deixam de ser interceptados por R_3 (porque sua MFT está *stale*) e atingem S. Desta forma, r_2 agora se conecta ao canal $<S, P>$ em S (Figure 2(c)). Algum tempo após *t2* irá expirar acarretando a retirada de r_1 das MFTs de S e R_3. Como R_3 pára de receber mensagens *tree*, sua MFT$<S>$ é destruída (Figura 2(d)). Agora, r_2 recebe os dados através do caminho mais curto a partir de S.

O roteamento assimétrico pode levar REUNITE a produzir cópias desnecessárias de pacotes em certos enlaces.[2] A Figura 3 mostra um exemplo. O primeiro receptor, r_1, envia um $join(S, r_1)$ que segue o caminho $r_1{>}R_4{>}R_2{>}R_1{>}S$. As mensagens $tree(S, r_1)$ seguem

[2] Esta possibilidade também exite para redes contendo roteadores puramente unicast ou quando um roteador REUNITE está sobrecarregado. Em ambos os casos, o nó de ramificação migrará para um roteador não ideal podendo acarretar a duplicação de pacotes. Consultar [4] para uma descrição detalhada.

a rota $S{>}R_1{>}R_6{>}R_4{>}r_1$. Suponha agora que r_2 se conecta e que o $join(S, r_2)$ passa por $r_2{>}R_5{>}R_3{>}R_1{>}S$. Os $tree(S, r_1)$ (produzidos por S) e os $tree(S, r_2)$ (criados em R_1) atravessam ambos o enlace R_1-R_6. Como R_6 não recebe mensagens $join$ destes receptores, ele não se identifica como nó de ramificação. S produz pacotes de dados endereçados a r_1, em seguida R_1 cria cópias endereçadas a r_2. Desta forma duas cópias de cada pacote atravessam o enlace R_1-R_6.

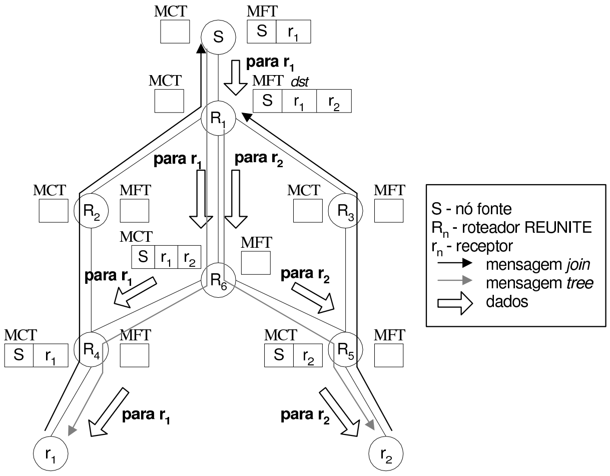
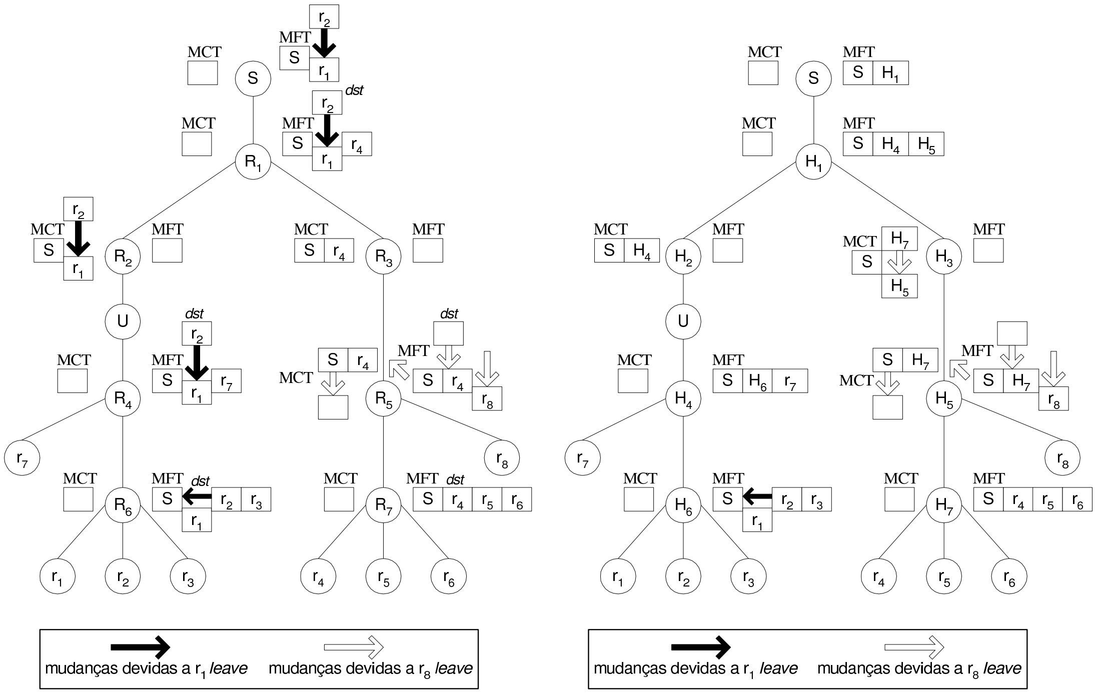


Figura 3: Duplicação de pacotes devido a rotas assimétricas no REUNITE.

Como conseqüência, o custo (número de cópias do mesmo pacote nos enlaces da rede) de uma árvore REUNITE pode ser maior que o custo de uma árvore por fonte (*source tree*) construída por um protocolo tradicional como PIM-SM (*Protocol Independent Multicast - Sparse Mode*)[7], uma vez que a técnica RPF (*Reverse Path Forwarding*) garante que no máximo uma cópia de cada pacote trafegará por cada enlace da rede. A seção a seguir descreve o funcionamento do protocolo HBH e mostra como ele lida com os problemas causados pelo roteamento unicast assimétrico.

3 O Protocolo HBH

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



(a) REUNITE. (b) HBH.

Figura 4: Comparação da reconfiguração da árvore após a saída de um membro.

A estrutura da árvore HBH possui a vantagem de maior estabilidade das entradas nas tabelas de roteamento que REUNITE. A contrapartida é que em HBH cada pacote recebido por um nó de ramificação produz $n+1$ cópias modificadas enquanto em REUNITE n cópias são produzidas. O mecanismo de gestão da árvore de HBH reduz o impacto da saída de um membro na estrutura da árvore. Isto é possível porque a entrada correspondente a um receptor é localizada o mais próximo possível deste receptor no HBH. Por exemplo, a saída de r_1 na Figura 4 possui um impacto maior na estrutura da árvore em REUNITE que em HBH. No pior caso, HBH precisará de uma mudança a mais que REUNITE (isto acontece quando um nó de ramificação se torna um nó de simples reenvio, por exemplo após a saída de r_8). No exemplo utilizado as rotas são simétricas, por isso não existe mudança de rota para os outros membros após a saída de um receptor. Isto pode ocorrer, no entanto, caso de r_2 no exemplo da Figura 2. Isto é evitado em HBH.

3.1 Gestão da árvore HBH

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

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

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

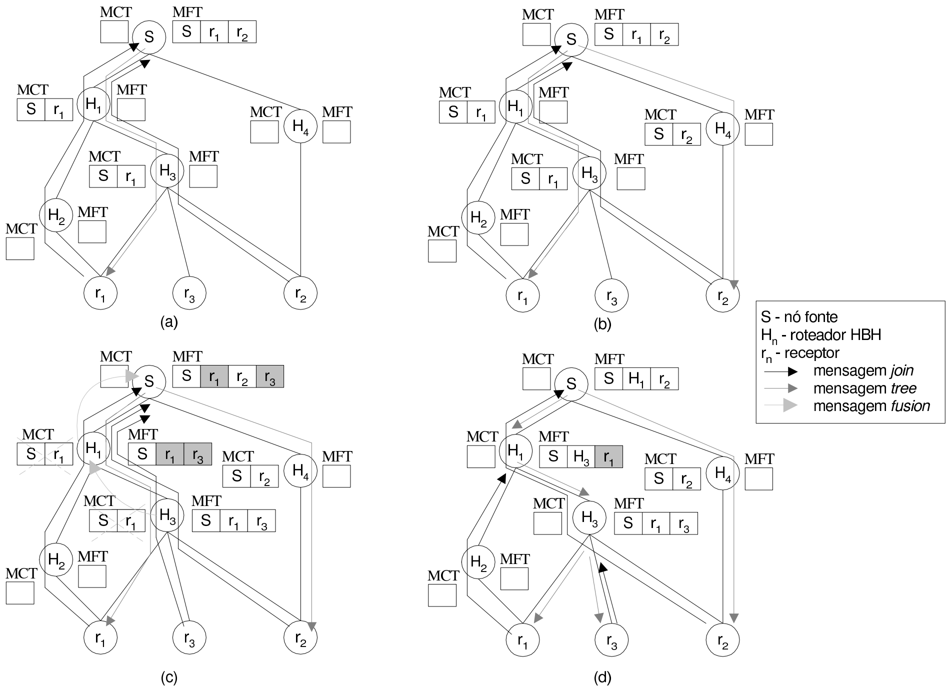


Figura 5: O mecanismo de construção da árvore HBH.

Considere novamente o primeiro exemplo da Seção 2.3 para mostrar a construção da árvore HBH. A Figura 5 retoma o cenário da Figura 2. r_1 se conecta ao canal em S, que começa o envio de mensagens $tree(S, r_1)$. Estas mensagens criam uma MCT$<S>$ contendo r_1 nos nós R_1 e R_3 (Figura 2(a)). Quando r_2 se conecta ao canal enviando o primeiro $join(S, r_2)$, este não é interceptado chegando a S (o primeiro $join$ nunca é interceptado). O $tree(S, r_2)$ produzido pela fonte cria uma MCT$<S>$ em R_4 (Figura 5(b)). Ambos os receptores estão conectados à fonte através do caminho mais curto (fonte receptor).

Suponha agora que r_3 (rotas unicast: $S>R_1>R_3>r_3$ e $r_3>R_3>R_1>S$) se conecta ao canal. r_3 envia um $join(S, r_3)$ para S, que começa a enviar mensagens $tree(S, r_3)$. Como R_1 recebe duas mensagens $tree$ diferentes, este envia um $fusion(S, r_1, r_3)$ na direção da fonte. A recepção do $fusion$ faz com que S *marque* as entradas r_1 e r_3 e inclua R_1 na MFT$<S>$. Da mesma forma que R_1, R_3 recebe os $tree(S, r_1)$ e $tree(S, r_3)$ e envia então uma mensagem $fusion(S, r_1, r_3)$ para a fonte (Figura 5(c)). A MFT de R_3 agora contém r_1 e r_3. Os $join(S, r_1)$ subseqüentes são interceptados por R_1 e atualizam a entrada r_1 (marcada) na MFT de R_1. Os $join(S, r_3)$ atualizam a entrada r_3 na MFT de R_3. A distribuição de dados se passa da seguinte forma. S envia pacotes endereçados a R_1, que os reenvia endereçados a R_3. R_3 cria cópias que são enviadas a r_1 e r_3. Subseqüentemente, como S deixa de receber os $join(S, r_1)$ e $join(S, r_3)$, as entradas correspondentes em sua MFT são destruídas. A estrutura estabilizada da árvore é mostrada na Figura 5(d). Desta forma, HBH utiliza o nó de ramificação ideal para a distribuição multicast. O problema apresentado na Figura 3 é resolvido através do envio de um $fusion(S, r_1, r_2)$ de R_6 para a fonte, de maneira equivalente ao exemplo desta seção.

4 Análise de Desempenho

O programa NS (*Network Simulator*)[11] foi utilizado na simulação do protocolo HBH. O objetivo dos experimentos é testar HBH e comparar a estrutura das árvores HBH e REUNITE. Para tanto, foram medidos o atraso médio entre a fonte e os receptores conectados ao canal assim como o número de cópias do mesmo pacote que os protocolos precisaram para cobrir todos os receptores.

4.1 Cenário de simulação

A Figura 6 apresenta a primeira topologia utilizada nas simulações. Esta topologia é típica de um grande provedor de serviços Internet (ISP)[12]. Sem perda de generalidade, considere que um receptor potencial é conectado a cada nó da rede. A existência de uma ou várias estações receptoras conectadas a um mesmo roteador de borda é mascarada pela utilização do protocolo IGMP (*Internet Group Management Protocol*)[13] e portanto não influencia o protocolo de roteamento. Por exemplo, na topologia ISP (Figura 6) os nós de 0 a 17 são roteadores e os nós de 18 a 35 são estações (receptores potenciais). Além desta topologia, foram realizados experimentos com uma topologia gerada aleatoriamente, maior (50 nós) e de maior conectividade (8.6 contra 3.3).

Dois custos, $c(n_1, n_2)$ e $c(n_2, n_1)$, são associados ao enlace n_1-n_2. O valor de c é um número inteiro aleatoriamente escolhido no intervalo $[1, 10]$. As simulações consideram um canal multicast onde um nó é fixado como fonte. Um número variável de receptores escolhidos aleatoriamente se conecta ao canal. Para cada tamanho de grupo foram realizadas 500 simulações. Os gráficos apresentam a média dos resultados obtidos.

4.2 Resultados

Além de HBH e REUNITE, foram analisados dois tipos de árvore construídas por protocolos multicast atuais. O NS possui um protocolo de roteamento multicast capaz de construir árvores compartilhadas e árvores por fonte de maneira semelhante ao protocolo PIM-SM [7]. A diferença é que a transição entre a árvore compartilhada e a árvore por fonte não pode ser feita de maneira automática como no PIM-SM, mas manualmente. Desta forma, “PIM-SM” nos resultados a seguir se refere a um protocolo que constrói apenas árvores compartilhadas e “PIM-SS” a um protocolo que constrói árvores por fonte.

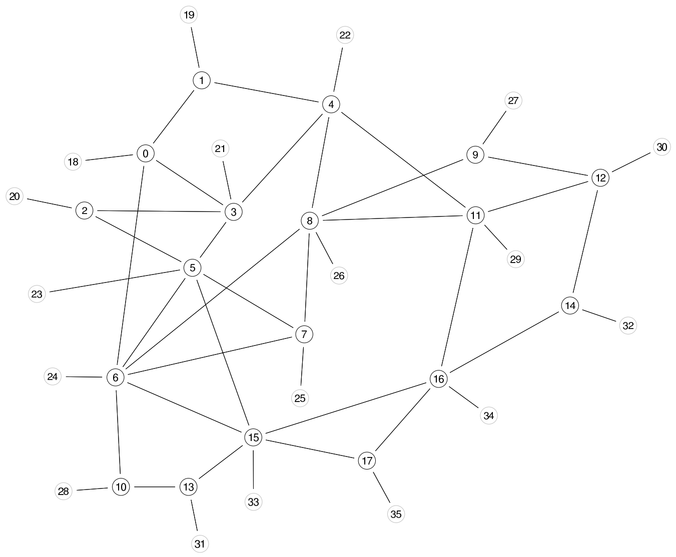


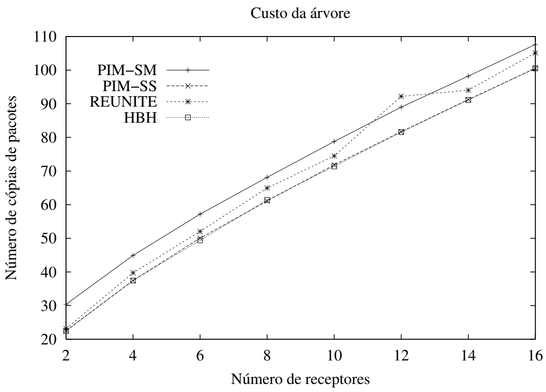
Figura 6: A topologia ISP.

A estrutura da árvore "PIM-SS" é portanto a mesma da construída por PIM-SSM (em padronização)[9] – uma árvore SPT reversa. Além de HBH, foi implementado um módulo REUNITE de acordo com [4]. Nos experimentos todos os roteadores são multicast.

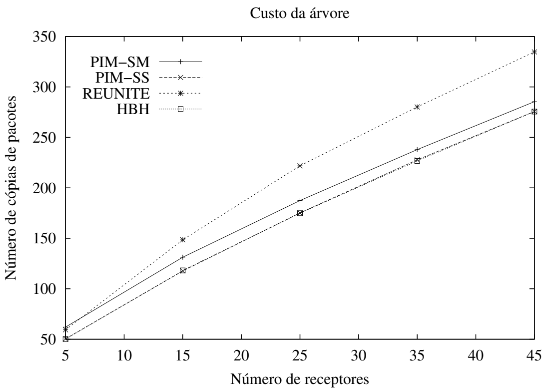
4.2.1 Custo das árvores construídas

A primeira análise se refere ao custo das árvores construídas. O custo da árvore é definido como o número de cópias do mesmo pacote transmitidos nos diversos enlaces da rede. Desta forma, o custo é diferente do número de enlaces presentes na árvore, uma vez que a técnica de unicast recursivo pode exigir a transmissão de mais de uma cópia do mesmo pacote sobre um mesmo enlace. Este fenômeno pode ser devido às assimetrias do roteamento unicast (Seção 2.3) mas também devido à presença de roteadores puramente unicast na rede, que não são capazes de atuar como nó de ramificação. No entanto, como nos experimentos realizados todos os roteadores são multicast, duplicatas do mesmo pacote sobre um enlace são sempre devidas ao roteamento assimétrico.

A Figura 7 mostra o custo médio das árvores construídas para diferentes tamanhos do grupo de receptores. Para a topologia ISP, o protocolo PIM-SM constrói as árvores de maior custo na maioria dos casos. Este resultado é esperado uma vez que este protocolo constrói árvores compartilhadas. Como o grupo multicast simulado é de 1 fonte para N receptores, o uso da árvore compartilhada é desvantajoso porque esta é centrada num ponto de *rendez-vous* (RP). A probabilidade de esta árvore ter maior custo que a árvore equivalente com raiz na fonte é alta. HBH e PIM-SS constroem as árvores de menor custo. Este comportamento também é esperado uma vez que PIM-SS constrói árvores por fonte e se baseia no algoritmo RPF. Por um lado, isto garante que no máximo uma cópia de cada pacote será transmitida em cada enlace e por outro lado que cada receptor é conectado à fonte através do caminho mais curto *reverso* (receptor fonte). O desempenho de HBH é similar porque cada receptor está conectado à fonte através do caminho mais curto (fonte receptor). O uso do caminho mais curto entre a fonte e o receptor ou entre o receptor e a fonte não altera o custo da árvore construída para estas topologias.



(a) Topologia ISP.



(b) Topologia de 50 nós.

Figura 7: Número de cópias médio de um mesmo pacote.

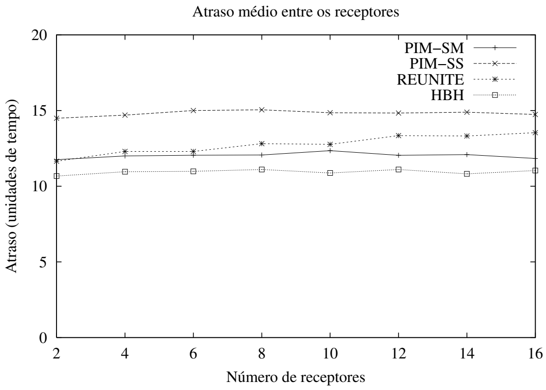
As curvas correspondentes a REUNITE na Figura 7 mostram que seu mecanismo de construção de árvores sofre efetivamente com os casos patológicos produzidos pelas assimetrias do roteamento unicast, como evocado na Seção 2.3. O fenômeno é menos freqüente quando o número de receptores é pequeno, uma vez que a probabilidade de dois receptores compartilharem um enlace na recepção de dados é pequena. Para a topologia ISP, o problema ocorre também com menos freqüência quando o tamanho do grupo é grande (a distribuição de receptores é densa) porque neste caso uma grande parte dos enlaces da rede estará sendo utilizada na árvore de distribuição. No entanto, este não é o caso para a topologia de 50 nós. Esta possui uma conectividade muito maior, o que significa que uma porcentagem menor dos enlaces da rede é utilizado, mesmo quando o tamanho do grupo é grande. Nesta topologia, a vantagem de HBH sobre REUNITE cresce com o tamanho do grupo. REUNITE também apresenta um desempenho pior que as árvores compartilhadas de PIM-SM como conseqüência dos nós de ramificação mal posicionados que levam à duplicação desnecessária de pacotes. A análise da curva obtida por HBH demonstra a eficiência do seu mecanismo de construção de árvores. A vantagem em termos de custo de HBH sobre REUNITE foi de 5% e 18% para as topologias ISP e de 50 nós, respectivamente, na média de todos os tamanhos de grupo. Pode-se concluir que o protocolo HBH proporciona economia de banda passante para redes assimétricas.

4.2.2 Atraso experimentado pelos receptores

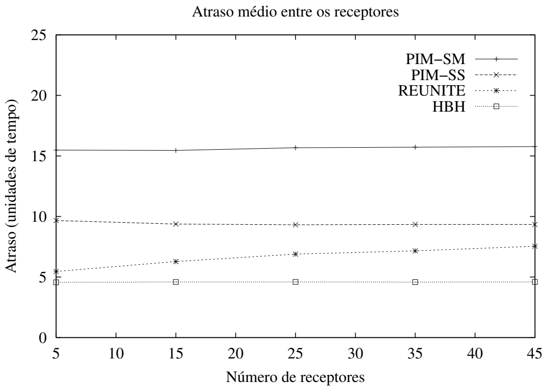
A Figura 8 apresenta o atraso médio experimentado pelos receptores do canal multicast para o mesmo conjunto de simulações. As curvas mostram que HBH é capaz de gerar rotas de menor atraso que REUNITE face ao roteamento unicast assimétrico.

A Figura 8(a) mostra dois resultados inesperados. Primeiro, PIM-SM apresenta um desempenho melhor que PIM-SS para a topologia ISP (embora PIM-SS construa árvores com a raiz na fonte enquanto PIM-SM constrói árvores compartilhadas). Este fenômeno é explicado pelo fato de PIM-SS construir árvores SPT *reversas* e não SPTs. Desta forma, o atraso não é minimizado. Por outro lado, na árvore compartilhada, os caminhos da fonte para cada receptor possuem todos uma porção em comum, o trecho entre a fonte e o ponto de *rendez-vous*. Como os dados são encapsulados em unicast entre a fonte e o

RP, o atraso é minimizado neste trecho. Como conseqüência, os caminhos fonte-receptor em PIM-SM possuem duas partes: entre a fonte e o RP onde o atraso é minimizado e entre o RP e o receptor onde o atraso não é minimizado uma vez que este trecho é um caminho mais curto *reverso*. Isto explica a vantagem de PIM-SM sobre PIM-SS para esta topologia. No entanto, o fenômeno não é observado na topologia de 50 nós, pois esta é maior e possui conectividade mais alta. Logo partir da fonte para o RP para após ir ao receptor gera, com grande probabilidade, um caminho mais longo que indo diretamente da fonte para o receptor. A segunda observação importante para a topologia ISP é que o efeito das assimetrias do roteamento sobre REUNITE pode ser forte o suficiente para levá-lo a apresentar um desempenho pior que PIM-SM (quando o grupo é grande).



(a) Topologia ISP.



(b) Topologia de 50 nós.

Figura 8: Atraso médio experimentado pelos receptores.

O desempenho de HBH é melhor que o de REUNITE para todos os tamanhos de grupo, em ambas as topologias. A vantagem sobre REUNITE cresce com o número de receptores, sendo de 14% em média para a topologia ISP. Para a topologia de 50 nós, os valores absolutos de atraso são menores devido à maior conectividade. No entanto, a vantagem de HBH é maior, de 30% em média. A maior diferença é também conseqüência da maior conectividade, que gera um maior número de possibilidades na construção da árvore – tornando o protocolo mais vulnerável às assimetrias do roteamento unicast.

5 Conclusões

Este artigo apresentou HBH, um protocolo de roteamento multicast que implementa a distribuição de dados através da técnica de unicast recursivo, originalmente proposta em REUNITE [4]. HBH permite a implantação progressiva do serviço multicast uma vez que é capaz de suportar roteadores puramente unicast de forma transparente. A concepção de HBH foi inspirada pelos protocolos REUNITE e EXPRESS, de forma a reunir os avanços destas propostas e contornar suas fraquezas. Os objetivos principais de HBH são:

- atravessar nuvens unicast;
- minimizar o impacto na estrutura da árvore devido à saída de um membro;
- construir árvores de menor custo nos casos onde o mecanismo de REUNITE falha;

- garantir que os receptores recebem os dados através do caminho mais curto a partir da fonte.

HBH possui um mecanismo original de gestão de árvores que se baseia em três mensagens. Os receptores periodicamente enviam mensagens *join* na direção da fonte. A fonte periodicamente envia mensagens *tree* em multicast sobre a árvore. Durante a viagem das mensagens *tree* através da árvore, nós intermediários podem gerar mensagens *fusion* que são responsáveis por refinar a estrutura da árvore.

HBH é capaz de construir árvores SPT mesmo na presença de roteamento unicast assimétrico. Além disso, HBH proporciona uma melhor utilização dos recursos da rede uma vez que minimiza a duplicação de pacotes na utilização do unicast recursivo. As árvores HBH são também melhor adaptadas a aplicações que não suportam grandes atrasos, como as aplicações interativas.

Os resultados de simulação mostram que o roteamento unicast assimétrico afeta o desempenho do protocolo de roteamento multicast. HBH mostrou-se uma alternativa promissora, uma vez que seu desempenho foi superior ao de REUNITE em termos de atraso e custo das árvores construídas. A vantagem cresce em redes maiores e de maior conectividade. Como trabalho futuro, a interface entre HBH e IP Multicast será detalhada e a possibilidade de incluir parâmetros de QoS na construção de árvores estudada.

Referências

[1] S. Deering, *Host Extensions for IP Multicasting.* RFC 1112, Aug. 1989.

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

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

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

[5] D. Waitzman, C. Partridge, and S. Deering, *Distance Vector Multicast Routing Protocol.* RFC 1075, Nov. 1988.

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

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

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

[9] S. Bhattacharyya, C. Diot, L. Giuliano, R. Rockell, J. Meylor, D. Meyer, and G. Shepherd, *A Framework for Source-Specific IP Multicast Deployment*, July 2000. Internet-draft: draft-bhattach-pim-ssm-00.txt.

[10] J. Moy, *Multicast Extensions to OSPF.* RFC 1584, Mar. 1994.

[11] K. Fall and K. Varadhan, *The ns Manual.* UC Berkeley, LBL, USC/ISI, and Xerox PARC, Jan. 2001. Available at http://www.isi.edu/nsnam/ns/ns-documentation.html.

[12] G. Apostolopoulos, R. Guerin, S. Kamat, and S. K. Tripathi, "Quality of service based routing: A performance perspective," in *ACM SIGCOMM'98*, pp. 17–28, Sept. 1998.

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

A Regras de Processamento de Mensagens HBH

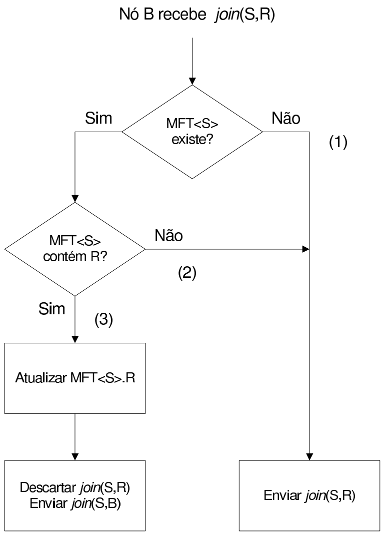
A Figura 9 apresenta as regras de processamento dos três tipos de mensagem HBH. Cada receptor, R, envia periodicamente uma mensagem $join(S, r)$ (em unicast) endereçada à fonte, S. S periodicamente envia (em multicast) um $tree$ sobre cada canal $<S, G>$.

Mensagem *join* (Figura 9(a)) - Quando um roteador, B, recebe um $join(S, R)$, ele o reenvia inalterado se B não possui MFT (1) ou se R não está presente na MFT de B (2). Somente se B possui uma entrada R em sua MFT, B intercepta o $join(S, R)$ e envia um $join(S, B)$ em seguida. Neste caso, B é um nó de ramificação do canal $<S, G>$ (3).

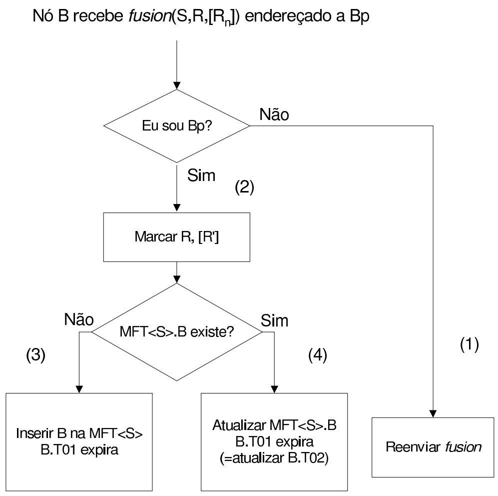
Mensagem *tree* (Figura 9(c)) - Uma mensagem $tree(S, R)$ recebida por um roteador, B, é tratada e reenviada em multicast. Se B é um nó de ramificação, B pode receber mensagens $tree$ endereçadas a B. Neste caso, B descarta esta mensagem e envia um $tree$ a cada nó presente em sua MFT (1). Se B é um nó de ramificação e a mensagem $tree(S, R)$ não é endereçada a B, existem duas possibilidades: R é um novo receptor (neste caso B insere R em sua MFT e envia uma mensagem $fusion$ na direção da fonte) (2) ou R está presente na MFT de B – o que significa que B não recebe as mensagens $join(S, R)$ enviadas por R – e neste caso B simplesmente atualiza a entrada R em sua MFT e envia à fonte uma mensagem $fusion$ (3). Se B não é um nó de ramificação, existem duas possibilidades: B não pertencia à árvore de distribuição de S e neste caso B cria uma MCT contendo R (4), ou B já estava na árvore de distribuição, porém não como um nó de ramificação (neste caso, B possui uma MCT$<S>$) (5). Se R está presente na MCT de B, nada mudou e B simplesmente atualiza MCT$<S>$ (6). Se R não está presente na MCT$<S>$, então pode ser que a MCT esteja *stale* e neste caso R substitui a antiga entrada da MCT (7), ou a MCT está em dia, o que significa que um novo receptor vai receber dados através de B e portanto B se torna um nó de ramificação. Isto implica a criação de uma MFT$<S>$, a destruição da MCT$<S>$ e o envio de uma mensagem $fusion$ na direção da fonte (8). Os $fusion$ produzidos por B contém uma lista de todos os nós que B mantém em sua MFT – nós para os quais B é nó de ramificação.

Mensagem *fusion* (Figura 9(b)) - Suponha que o nó B recebe um $fusion(S, R, ...R_n)$ do nó B_p. Se a mensagem não é endereçada a B, então B simplesmente a reenvia na direção da fonte (1). Se a mensagem é endereçada a B, então B *marca* as entradas na MFT$<S>$ correspondentes aos receptores listados na mensagem $fusion$ (2). B_p é adicionado à MFT de B, caso não estivesse presente. Além disso, o temporizador *t1* correspondente a B_p é expirado – desta forma B_p torna-se *stale* (3). Conseqüentemente, B_p será usado para o reenvio de dados, mas não na replicação de mensagens $tree$. Se por outro lado B_p já estava presente na MFT de B, então o temporizador *t2* correspondente a B_p é atualizado (o que evita a destruição da entrada B_p), porém *t1* é mantido expirado (4).

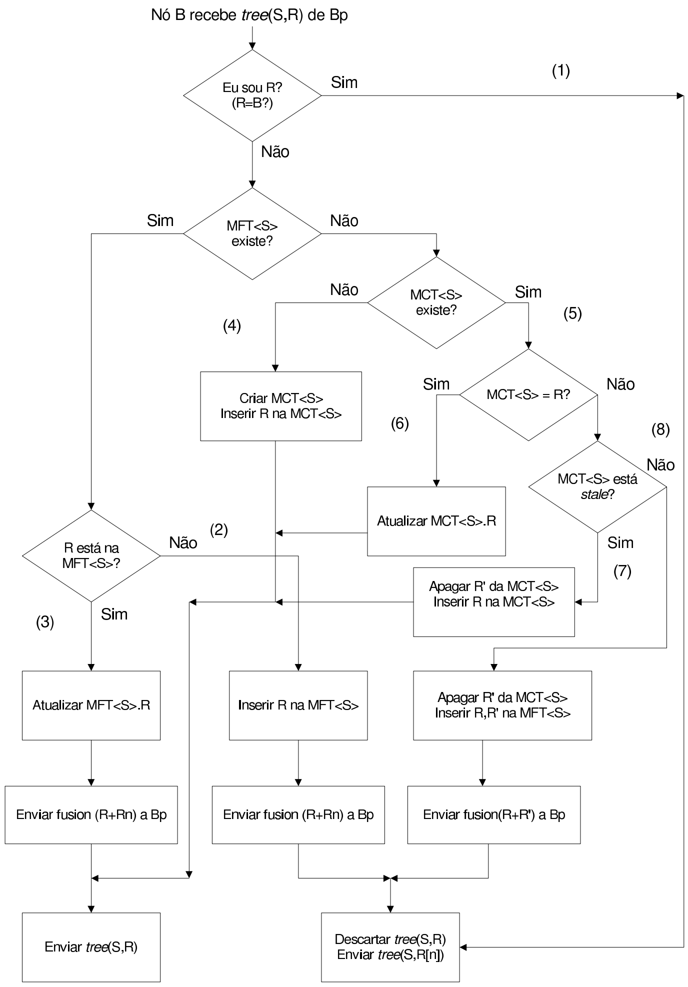
Se em seguida B_p (o nó que produziu a mensagem $fusion$ incluindo $R, ...R_n$) recebe mensagens $join$ produzidas por algum dos receptores $R, ...R_n$, B_p as intercepta e envia um $join(S, B_p)$ na direção da fonte. Neste caso os temporizadores correspondentes às entradas $R, ...R_n$ na MFT de B irão eventualmente expirar causando a destruição das entradas, enquanto a entrada B_p será atualizada pelos $join(S, B_p)$. Se B_p não receber as mensagens $join$ de algum receptor, R_i, entre os $R, ...R_n$, a emissão de mensagens $fusion$ deve continuar uma vez que é outro nó, localizado mais alto na árvore, que recebe os $join(S, R_i)$ e periodicamente produzirá as mensagens $tree(S, R_i)$. No entanto, este outro nó não é responsável pelo envio de dados a R_i, mas B_p ao invés disso. Desta forma HBH é capaz de lidar com o segundo problema devido ao roteamento assimétrico da Seção 2.3.



(a) Mensagem *join*.



(b) Mensagem *fusion*.



(c) Mensagem *tree*.

Figura 9: Processamento de mensagens HBH.