按需分枝组播*

金志权+, 项晓晶, 陈佩佩

(南京大学 计算机软件新技术国家重点实验室,江苏 南京 210093)
(南京大学 计算机科学与技术系,江苏 南京 210093)

On-Demand Branching Multicast

JIN Zhi-Quan+, XIANG Xiao-Jing, CHEN Pei-Pei

(State Key Laboratory for Software Novel Technology, Nanjing University, Nanjing 210093, China)
(Department of Computer Science and Technology, Nanjing University, Nanjing 210093, China)
+ Corresponding author: Phn: 86-25-3594863, E-mail: jinzq@public1.ptt.js.cn
http://www.nju.edu.cn
Received 2001-10-17; Accepted 2001-12-27

Jin ZQ, Xiang XJ, Chen PP. On-Demand branching multicast. *Journal of Software*, 2003,14(3):553~561.

Abstract: After analyzing and summarizing the main current research outcomes of IP multicast routing, a multicast routing algorithm called on-demand branching multicast is proposed. On-Demand branching multicast uses a new multicast tree maintenance method, in which the tree is maintained only by partial nodes or key nodes. That is different from the traditional methods in which the tree is maintained by all the nodes on the tree, and so as to save the Internet resources.
Key words: multicast routing; multicast distributed tree; multicast model; coexistence and convertibility

摘 要: 在分析总结目前IP组播路由研究的主要成果基础上,提出了一个新的组播路由方案,按需分枝组播.它采用了一种全新的组播树维护方式,即组播树由树上的部分节点(关键节点)维护,不同于现有的组播树由所有树上节点维护的方式,从而节省了网络资源.
关键词: 组播路由;组播分布树;组播模型;共存性和互转换性
中图法分类号: TP393 **文献标识码**: A

1 现有组播路由的缺点

组播路由建立在组播分布树的基础上.组播数据包沿着组播分布树传送给组内所有成员.现有组播路由协议的组播分布树是依靠组播树所经由的每一跳路由器共同合作来维护的[1~7].例如在图1中,组播分布树要依靠路由器R1~R9的共同维护,这些路由器都要记录该组播树的相应信息,对于单向组播树来说,这些信息包括组标识、进入接口、发出接口等.同时,这些路由器还要加入到组播树的维护过程中.对于密集模式的组播路由协议,

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2001AA110235 (国家高技术研究发展计划)

第一作者简介: 金志权(1941－),男,江苏吴县人,教授,主要研究领域为分布式并行计算,计算机网络及安全.

如 DVMRP[4]和 PIM-DM[8],采用一种扩散和剪枝的方式,每隔一段时间,原有的组播树就会失效,组播信息流扩散到网络中,接着没有组成员的路由器开始剪枝过程,使组播信息只流向需要的节点.此时,不仅组播树上的路由器,非组播树上的路由器也都要参与到组播树的维护中.对于稀疏模式的组播路由协议,如 PIM-SM[9]和 CBT[10,11],采用共享树,使用显式的加入机制来建立组播分枝,分枝上的所有路由器都要维护分枝信息,为了防止分枝的失效,需要组播分枝上的路由器采用周期性的更新机制.例如在 PIM-SM 中,通过向共享树周期性地重新发送加入信息,分枝上的路由器根据加入确认信息周期性地刷新分枝信息.剪枝过程同样需要被剪分枝上的所有路由器的参与.例如在图 1 中的路由器 R2,R3,R6 和 R7,在它们直接相连的链路上没有组成员,而且因为它们不在组播树的分枝点处,不需要负责组播数据包的复制,因此它们在组播树中仅仅是起到中间路由的作用,但却要维护组播树信息,并且要参加到频繁地组播树的维护过程中.如果源节点和组成员相距较远,中间会经过许多这样的路由器,特别是网络中同时会存在很多的组播组,这些路由器将会位于许多组播树的路径上,这样会消耗很多网络资源,给这些路由器增加许多不必要的负担,极大地影响了网络数据传输的性能.

由于现有组播树是通过树上的所有路由器共同维护的,因此对于网络拓扑变化的适应性不强,只要其中一个路由器失效,则所有通过此路由器接收组播信息的节点都要重新开始一次组加入过程.

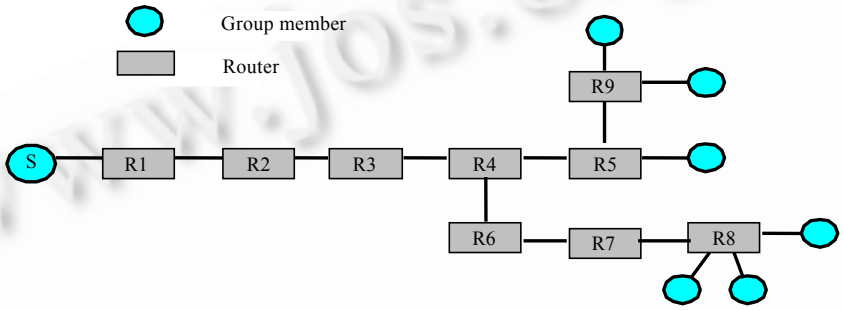

Fig.1 A multicast tree sample

图 1 组播树示例

2 ODBM 解决方式

本文提出按需分枝组播(on-demand branching multicast,简称 ODBM).ODBM 采用一种全新的组播树维护方式,即只有处于组播树分枝点处的路由器和本地链路上有组成员的路由器需要保存组播树的有关信息,并参加组播树的维护过程.组播树上的其他路由器只是以普通单播的路由方式组播数据包,无须维护组播树的任何信息.

这种解决方式克服了现有组播路由协议中组播树维护方法中存在的缺陷,减轻了沿途路由器的负担,优化了网络数据传输的性能,并对拓扑变化有一定的适应性.ODBM 的另一个主要优点是,克服了现有组播路由的扩展性能不佳的问题.现有组播路由的组播树维护方式基于一种平面的网络结构,因此缺乏可扩展性,通常工作在 M-Bone,无法适用于整个因特网范围内的组播,因此开发可以扩展到因特网范围的组播路由是目前组播研究的一个重点[6].目前的单播路由结合因特网的层次型体系结构,采用分层的路由方式,因此具有很好的可扩展性.本算法将组播树的维护局限在树中的关键节点,其他节点直接利用了现有的单播路由传输数据包,因此也就具有了与单播路由相似的可扩展性能.

2.1 ODBM组播模型

2.1.1 模型构成实体

S:组播组的源节点主机.

DR(designate router):指定路由器.位于与源主机 S 直接相连的链路上,负责 S 发出的组播数据包路由的指定路由器.如图 1 中的 R1.

LR(leaf router):叶子路由器.与该路由器直接相连的链路上有组成员,且该路由器是组播树树枝的终点.如图 1 中的路由器 R8 和 R9.

BR(branching router):分枝路由器.该路由器位于组播树的分枝点处,此时与该路由器直接相连的链路上可以存在组成员.如图 1 中的 R4 即为分枝路由器.或者该路由器位于组播树上,且与其直接相连的链路上存在组成员,如图 1 中的 R5.

GM(group member):组成员主机.

其中,S,DR,LR 和 BR 要共同负责组播树的维护.DR,LR 和 BR 称作组播树的关键路由器,记为 KR(key router).

模型所用数据结构

在每个 KR 中都保存有相应的 ODBM 路由表,表中至少包含有以下这些信息:组播树标识(S,G);上游 KR 的 IP 地址,组播数据包的进入接口,为防止路由循环,必须保证只有一个上游 KR 表项;下游 KR 的 IP 地址与相应接口,该接口是通过查找单播路由表获得的;如果本地链路上有(S,G)的成员,还要包含本地接口,并标明本地;另外,各表项还要标明有效时间,用于组播树的维护.

组播树上的其他非 KR 路由器和非组播树上的路由器,不包含 ODBM 路由表或 ODBM 路由表(S,G)表项为空.

为叙述简便,下面使用一些符号来表示交换信息和节点.(S,G)表示组 G 中以源 S 为根的组播树.各个节点除了上述的 S,DR,LR,BR 和 KR 外,路由器以 R 表示.在相同符号后用阿拉伯数字加以区分,例如 KR1,KR2.交换信息以信息源节点_信息目的节点_信息名称(参数表)的格式来表示,例如 R1_S_JoinRequest(S,G)为 R1 发给 S 的请求加入组(S,G)的信息.主要信息类型汇总可参见第 2.2 节.

2.1.2 组播树的建立

ODBM 是一种稀疏模式的组播路由模型,它采用稀疏模式的按需建立组播分枝的方式,即使用显式加入的模式.对于组播组成员分布较密的密集模式而言,也可以使用本组播算法,但由于此时在组播树中关键路由器所占的比重很大,使用本算法并没有产生明显的节省网络资源的效果.因此本算法适用于组成员分布较稀疏、关键路由器在组播树中所占比例较小的情况.

组成员的维护

各个路由器采用 IGMP 协议(IPv4)或者 ICMPv6 和 MLD 协议(IPv6)[4,5]来维护本链路上的组成员,并根据本链路上某个组的组成员的有无采取相应的加入组与退出组的动作.

组播树分枝的建立

路由器 R1 收到本地链路上的主机请求加入(S,G)的信息,如果此时它不是(S,G)上的 KR,即它没有保存该组播树的信息,R1 将创建 ODBM 路由表(S,G)表项,表中填写本地需要组播信息的相应接口号,同时向源 S 发送加入请求信息 R1_S_JoinRequest(S,G).此信息沿着向 S 的方向传送,在传送过程中遇到组播树上的 KR1,KR1 接收此信息,并开始建立抵达 R1 的组播树分枝并向 R1 确认加入的过程,KR1 首先从单播路由表中查找抵达 R1 的接口号 IR,然后查找 ODBM 路由表的(S,G)表项.查找结果分为下面 3 种情况:

(1) IR 与上游接口号相同.

为了防止路由循环,简单地丢弃此请求信息.

(2) IR 与某个下游接口号相同.

设该下游接口号对应的下游 KR 为 KR2.KR1 向 R1 的方向发送一个分枝请求信息 KR1_R1_BranchRequest(R1,KR2),信息里包含 R1 和 KR2 的 IP 地址.信息沿 IR 接口发送出去,设收到此信息的下一跳路由器为 R2.R2 从单播路由表中分别查找通往 R1 和 KR2 的接口号,分两种情况:

(Ⅰ) 若接口号相同,则继续将此信息往下传送,之后收到此信息的路由器都作类似于 R2 的处理.

(Ⅱ) 若接口号不同,它向 KR1 返回一个分枝确定信息,包含自己的 IP 地址,R2_KR1_BranchConfirm(R2).然后 R2 查找 ODBM 路由表(S,G)表项,若此表项不存在,则建立表项并填入相关信息,以 KR1 为上游 KR,R1 和 KR2 为下游 KR,同时向 R1 和 KR2 发送加入确认信息 R2_R1_JoinConfirm(R2)和 R2_KR2_JoinConfirm(R2);若

此表项已经存在,即 R2 是此组播树上的 KR,R2 将采取与 KR1 相同的方式将通往 R1 和 KR2 的端口号与表项中的下游端口号进行比较并作类似处理.KR2 收到确认信息后将 ODBM 路由表中的上游 KR 改成确认信息中携带的 IP 地址所对应的 KR.

KR1 收到分枝确认信息后,将接口号 IR 对应的下游 KR 的 IP 地址改为确认信息中包含的 IP 地址.

(3) 没有与 IR 相同的接口号.

在 ODBM 路由表(S,G)表项中新建一个下游 KR 表项,填入 R1 的 IP 地址和接口 IR.同时,直接向 R1 发送一个加入确认信息 KR1_R1_JoinConfirm(KR1),包含自己的 IP 地址.

R1 收到确认信息之后,在 ODBM 路由表(S,G)表项中创建上游 KR 表项,以确认信息中携带的 IP 地址作为上游 KR 的地址,以确认信息进入的接口作为上游接口.若未收到确认信息,则采取一定的重发策略.

例如,图 2 中的路由器 R5 发现本地有组(S,G)的成员,它向源主机发送请求加入组信息,假设该信息沿图中虚线的方向流动.此信息被途中遇到的第 1 个组播树上的 KR,即图中的 BR1 接收,假设从 BR1 到 R5 的单播路径为图中虚线箭头的反向,BR1 查找单播路由表和 ODBM 路由表的结果将是上述(2)的情况,即通往 R5 的单播路由的接口与通往下游 KR(LR1)的接口相同,则 BR1 从此接口发出分枝请求信息,信息被传到 R3 处,R3 处的情况如(2)中的(Ⅰ)所述,R3 将直接将信息往下传递.信息包又被传到 R4 处,R4 处的将是(2)中(Ⅱ)所述的情况, R4 向 BR1 传回一个分枝确认信息.BR1 对 ODBM 路由表作相应修改,使下游 KR 中的 LR1 被替换为 R4.R4 也对自己的 ODBM 路由表进行修改,以 BR1 作为上游 KR,R5 和 LR1 作为下游 KR,同时向 R5 和 LR1 发出加入确认信息.R5 和 LR1 收到此信息后,将上游 KR 设置为 R4.这样,R4 变为组播树中的 KR,并建立了一个到达 R5 的新的组播分枝,组播树中的其他 KR 仍然保持正确的上游和下游 KR 的信息,共同维护一个正确的 ODBM 组播树.

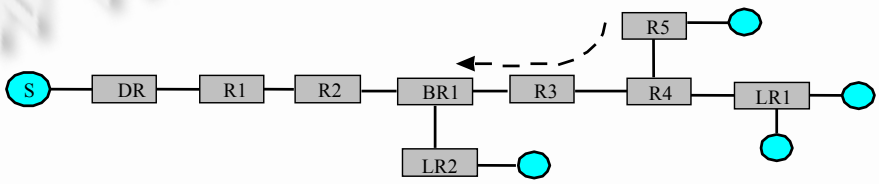

Fig.2 Branch creation of ODBM multicast tree

图 2 ODBM 组播树分枝的建立

2.1.3 组播树的剪枝

若某一个 KR(设为 KR1)发现自己的下游分枝或本地组播分枝已经失效,该失效可能是因为下游 KR 向它发出剪枝信息或者本地已没有组播组成员引起的,则 KR1 删除 ODBM 路由表中对应此分枝的表项,并检查 ODBM 路由表中对应(S,G)的表项,有下列几种情况:

(1) 若下游 KR 列表为空,且本地接口列表也为空.KR1 要向它的上游 KR(设为 KR2)发送剪枝信息 KR1_KR2_UnBranch(KR1),包含自己的 IP 地址,同时 KR1 也转变为非关键路由器.

(2) 仅剩一个下游 KR 表项,且本地接口列表为空.此时 KR1 在组播树中仅起到传输的作用,应取消其 KR 的功能.此时,它向上游的 KR2 发出一个 KR1_KR2_Update(KR1,KR3)信息,KR3 为 KR1 的下游 KR,KR2 收到此信息后,将下游 KR 列表中的 KR1 替换为 KR3;同时 KR1 向下游的 KR3 发送一个 KR1_KR3_JoinConfirm(KR2)信息,KR3 于是将上游 KR 改为 KR2.

(3) 其他情况.KR1 不发送任何信息.

上游 KR 若收到 KR1 发来的剪枝信息,删除 KR1 对应的下游 KR 表项,并作与 KR1 类似的处理.

2.1.4 组播树的维护

为保持组播树的有效性并适应网络拓扑的变化,必须对组播树进行维护.此维护过程仅需由组播树上的 KR 参加.各 KR 的 ODBM 的路由表的各表项保存有相应的有效时间值.各个 KR 周期性地向其上游和下游 KR 发送有效信息,指明自己的有效性,此周期的时间间隔必须小于有效时间值.如果某 KR(设为 KR1)中的 ODBM 路由表中的一个表项的有效时间快要结束,但仍未收到相应 KR 的有效信息,它就向该 KR 发送探询信息,并采用一定的重发策略,如果收到返回的有效信息,KR1 就将相应表项的有效时间重置;如果有效时间已过,但仍未

收到有效信息,则删除相应表项,此时分为两种情况:

(1) 如果被删表项对应的是上游 KR,说明此组播树枝已不能路由源 S 发出的组播数据包,即 KR1 为根的组播树枝已经失效.KR1 将向所有的下游 KR 发送失效信息,并删除所有上游 KR 和下游 KR 的表项.下游 KR 收到此失效信息后,删除上游 KR(即 KR1)对应的表项,并采取本步骤所述的和 KR1 相同的措施.接着被删除的以 KR1 为根的组播树树枝上(包括 KR1)的所有本地链路有(S,G)组成员的路由器重新开始组播组的加入过程,重新建立组播分枝.

(2) 如果被删表项对应的是下游 KR 或本地接口,则采取如第 2.1.3 节组播树剪枝中所述的类似措施.

2.1.5 组播数据包的路由

组播数据包将从源 S 沿着组播树向下传送.组播树上的 KR 根据 ODBM 路由表进行组播数据包的复制与转发.组播树上的非 KR 按照单播路由转发组播数据包.

移动 IPv6 中使用归宿地址目的选项,使移动节点对转交地址的使用对通信伙伴透明.受到此方法的启发,本组播算法也使用一个组播地址目的选项,达到 ODBM 组播路由对非 KR 和接收节点透明的目的.根据 IPv6 的目的选项的定义,该目的选项只被 IP 数据包的目的地址域所指明的目的节点来处理.该组播地址目的选项的格式如图 3 所示.

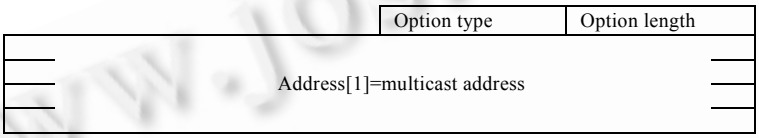

Fig.3 Destination option format for multicast

图 3 组播地址目的选项格式

S 发出的组播数据包的格式是以 S 为源节点,DR 为目的节点,同时携带包含有组地址的组播地址目的选项.DR 收到此数据包后发现存在组播地址目的选项,于是知道要采用 ODBM 方法传送此组播数据包.为了防止路由循环,它首先检查数据包进入的接口是否等于 ODBM 路由表中的上游接口,若不等,则抛弃数据包;若相等,则检查每一个下游 KR,为每一个下游 KR 复制一个组播数据包,并将该数据包的目的地址改成此 KR 的 IP 地址,再沿相应接口发送出去.接着,沿途的路由器将按照数据包的目的地址(即下游 KR 的 IP 地址),将数据包单播至此下游 KR.下游 KR 收到组播数据包后采取与 DR 相似的处理方法,首先检查进入接口防止路由循环,接着向每个下游 KR 发送组播数据包.如果 ODBM 路由表中(S,G)表项有本地接口表项.还要向本地发送组播数据包.当组播组的成员接收到传递来的数据包之后,发现组播地址目的选项,于是知道是该组播组的数据包,并可以将组播地址替换数据包中的目的 IP 地址,使 ODBM 算法的使用对上层透明.

3 算法实现

3.1 实现环境

为了证明 ODBM 算法的正确性和可行性,对该算法进行了模拟实现.实现环境采用了 UCLA 大学提供的 PARSEC 并行模拟环境[12,13].PARSEC 是一种基于 C 的离散事件的模拟语言,可用来模拟物理实体与其间的信息交换.

3.2 主要模块与功能

entity driver(int argc,char **argv)

代码的总控模块.主要功能包括读入并建立网络拓扑结构,对各个路由器的行为进行总控.

entity sender(ADDRESS group,ADDRESS destno,ename Driver,ename routers[N])

用于模拟组播组的数据源,向组播组发送组播数据包.

entity router(int myid,ename Driver)

算法的主要实现模块,模拟路由器.对路由器接收到的各个信息进行处理并采取相应动作.

int sendJoinRequest(ADDRESS source,ADDRESS dest,ADDRESS group,ADDRESS srcJoin,ADDRESS router,ename routers[N])

int sendJoinConfirm(ADDRESS source,ADDRESS dest,ADDRESS group,ADDRESS srcJoinCfm,ADDRESS router,ename routers[N])

各个信息发送模块,由路由器模块调用发送相应的控制信息.代码的主要模块之间的关系如图 4 所示.

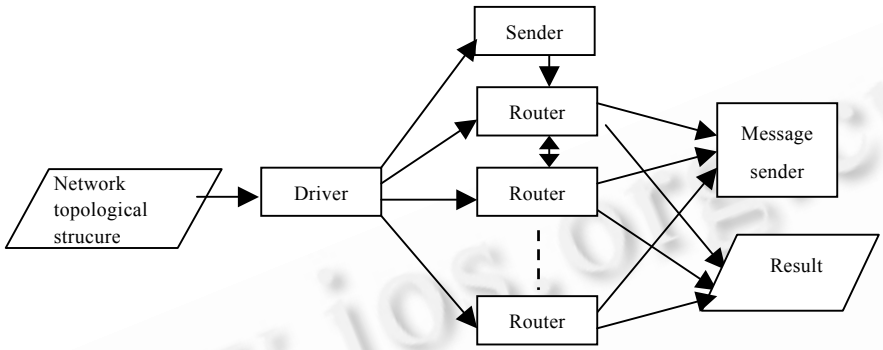

Fig.4 Relations of the main modules

图 4 代码的主要模块之间的关系

3.3 主要信息类型

表 1 列出了代码中使用的主要控制信息的名称、参数与作用.

Table 1 Control message table

表 1 控制信息列表

Message name	Parameters	Fuction
JoinRequest	S,G, R sending request	Join request message
JoinConfirm	S,G, up stream KR	Join confirm message
BranchRequest	S,G, branch vertex R and KR	Branch request message
BranchConfirm	S,G, down stream KR, branch vertex KR	Branch confirm message
UnBranch	S,G, R sending request	Unbranch request message
UpdateDS	S,G, old down stream KR, new down stream KR	Message for updating down stream KR
TransOld	S,G, up stream KR, down stream KR, root R	Control message for transforming ODBM into existing models
TransOldCfm	S,G, down stream KR, root R	Transform confirm message
TransODBM	S,G, up stream KR, root R	Control message for transforming existing models into ODBM
TransOdbmCfm	S,G, down stream KR, root R	Transform confirm message
TransEnd	S,G	Transform end message
Packet	S,G, sequence number	Multicast packet for testing
BeginMG	G	
BeginJoinSG	S,G	
FinishJoin	R	
BeginUnBranch	S,G	
BeginTransOdbm	S,G	Messages used by main module and other modules
BeginTransOld	S,G	
BeginSend	Driver	
FinishSend	Sender	
PrintOdbmTb	Next R	
PrintMcastTb	Next R	

3.4 主要数据结构

(1) ODBM 组播路由表

typedef struct down_addr{

```
        ADDRESS    down_stream;
        ADDRESS    down_interface;
        int    status;                    /*used for the transfer between old and odbm tree*/
        struct down_addr    *next;
}DOWNADDR;
typedef struct odbm_table{
        ADDRESS         group;            /*Group Address*/
        ADDRESS         source;           /*Source Address*/
        ADDRESS    up_stream;             /*The IP Address of up_stream Key Router*/
        ADDRESS    up_interface;          /*The interface of the up_stream*/
        DOWNADDR *down_list;              /*The list of the down_stream KR and interface*/
        LOCALADDR *local_list;            /*The list of the local receiver*/
        struct odbm_table *next;          /*The odbm_table items for other source-group*/
} ODBM_TABLE;
```

(2) 现有组播路由表

```
typedef struct mcast_table{
        ADDRESS    group;                 /*Group Address*/
        ADDRESS         source;           /*Source Address*/
        ADDRESS    up_interface;          /*The interface of the up_stream*/
        DOWNIF     *down_list;            /*The list of down interface*/
        struct mcast_table *next;         /*The old mcast_table items for other source-group*/
}MCAST_TABLE;
```

4 与现有组播模型的互操作

4.1 与现有组播模型的共存

ODBM 组播模型的一个优点就是与现有组播模型的共存性.一棵组播树可以同时由本组播模型和现有组播模型维护.组播树的不同部分可以根据需要采用其中一种组播树来维护的方式.

例如图 5 中的组播树,在区域 A 和区域 B 中的组播组成员分布较密集,可以按照现有组播树的方式建立以 RA 和 RB 为根的子树,即子树由树上的所有路由器负责维护.RA 和 RB 加入到以 S 为根的 ODBM 组播树上,并成为树上的 KR,同时在它们的 ODBM 路由表的下游 KR 表项中注明采用现有组播方式.当 ODBM 组播数据包传递到 RA 和 RB 时,它们从 ODBM 路由表中获知下游使用现有的组播树,于是将组播地址目的选项中携带的组播地址替换 ODBM 组播数据包中的目的节点 IP 地址,并去除此组播地址目的选项,产生现有的组播数据包.RA 和 RB 检查现有组播路由表获知下游发出接口,将产生的组播数据包复制,从这些接口发送出去.然后,组播数据包将沿着以 RA 和 RB 为根的现有组播树按照现有组播路由来传递.

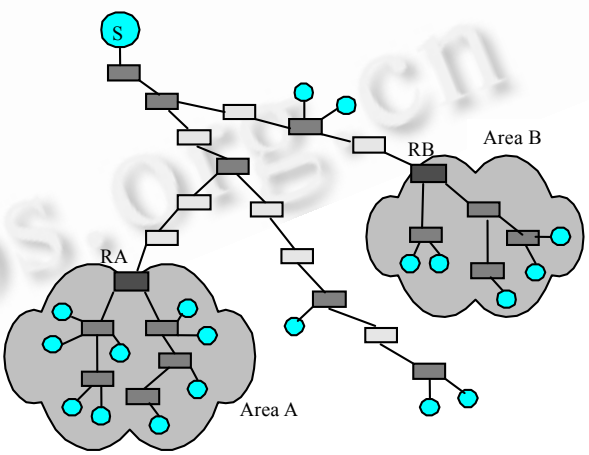

Fig.5 Two multicast models coexistence
图 5 两种组播模型的共存

4.2 与现有组播模型的互相转换

ODBM 适用于组播成员较稀疏、组播树上 KR 与非 KR 的比例较小的情况.可以在组播树上采取一定的监控措施,例如由源节点 S 发出探测包,计算 KR 与非 KR 的大致比例,当此比例超过预定的阈值时,可以发起转换

过程,由 ODBM 转换为现有的组播树.若小于预定的阈值,也可以由现有的组播树转换到 ODBM 组播树.

4.2.1 ODBM 到现有组播树的转换

源节点按照 ODBM 方式向下游 KR 发送 TransOld(S,G)信息(参见第 3.2 节),各 KR 收到此信息后,建立现有的组播路由表,将 ODBM 组播路由表中的组标识、上游接口和下游接口填写入该表.然后向下游 KR 传递 TransOld(S,G)信息,传递此信息的途中非 KR 同时也建立现有的组播路由表,将信息包的进入接口和发出接口作为上游接口和下游接口.当 LR 收到此信息包后,除了建立现有的组播路由表以外,向上游 KR 发送 TransOldCfm(S,G)信息.上游 KR 收到此信息后,在 ODBM 路由表中该 LR 对应的表项上作标记,当 KR 的所有下游 KR 都发来了 TransOldCfm(S,G)信息,它向自己的上游 KR 发送 TransOldCfm(S,G)信息.当源 S 收到所有下游 KR 的确认信息后,将删除或清空(S,G)对应的 ODBM 表项,并按照现有组播方式发送组播数据包.然后,数据包将沿着刚建立的现有组播树传递,发送中遇到的原 KR 发现数据包已按照现有方式发送,则删除或清空(S,G)对应的 ODBM 表项.至此整个转换过程结束.

此方式不仅可用于整个 ODBM 组播树的转换,还可用于组播树中的部分子树的转换.此时,转换过程由子树的根发起,当整个子树转换完毕之后,由子树的根负责将 ODBM 组播树上发来的数据包转换为现有组播数据包沿子树向下发送.结果就形成了第 4.1 节中的两种组播模式共存的情况.

4.2.2 现有组播树到 ODBM 的转换

此过程是上一个过程的逆过程.由源 S 发出 TransODBM(S,G)信息,DR 收到此信息后建立 ODBM 路由表,将此信息包的源节点(即 S)IP 地址作为上游 KR 的 IP 地址,对应的上游接口从现有组播路由表中获取,接着将此信息的源节点 IP 地址改为自己的 IP 地址并往下传递,树上的路由器收到此信息后检查组播路由表,如果只有一个下游接口,继续将此信息往下传递;如果不只一个下游接口,按照 DR 的方式建立 ODBM 组播路由表,并向下发送 TransODBM(S,G)信息;当 LR 收到此信息后,建立 ODBM 组播路由表,并返回 TransODBMCfm(S,G)信息,上游 KR 收到此信息后,将此信息包的源节点(即 LR)IP 地址填入 ODBM 组播路由表的相应下游 KR 表项中,并从单播路由表中获取相应的下游接口,当所有下游 KR 都发来确认信息后,它向上游 KR 发送确认信息.当 S 收到确认信息后,它沿现有组播树向下发送 TransEnd(S,G)信息,接着按 ODBM 发送组播数据包.树上的各节点收到 TransEnd(S,G)信息后,删除或清空对应(S,G)的现有组播路由表的表项.整个转换过程结束.

与第 4.2.1 节一样,此转换过程也很灵活,可以用于部分子树的转换.另外一种情况是,当组播树需要转换为 ODBM 模式,某子树由于组成员分布较密,准备继续使用现有模式时,该子树的根不向下发送 TransODBM (S,G)信息,而是直接向上游 KR 返回一个 TransODBMCfm(S,G)信息,这样,该子树将继续保持现有组播方式,与组播树其他部分的 ODBM 方式共存.

5 小 结

我们用如图 6 所示的网络拓扑结构对 ODMB 维护算法以其与传统组播树的共存和互相转换进行了验证性测试,都得到了正确的结果.

组播树的维护要花费路由器各种资源.对于密集模式的组播路由,如 DVMRP[2]和 PIM-DM[8],所有路由器都要周期性地扩散组播信息,并进行必要的剪枝,从而花费了非组播树上路由器的大量资源.对于稀疏模式的组播路由,如 PIM-SM[9,14]和 CBT[10,11],用显式的加入机制建立组播分枝(剪枝过程也一样)仍需要分枝上的路由器周期性地重新发送加入信息来维护.大量中间路由器的参与无疑是一种资源的浪费.尤其需要指出的是,通常工作在 M-Bone 的算法无法适用于 Internet[6],把一些算法简单地从 M-Bone 搬到 Internet 上将会产生网络资源的大量浪费.因此,当一个组跨越地理上的很大范围时,大量路由器必须从与它们无直接关系的组播中解放出来.为此,我们提出了 ODBM.ODBM 只有关键路由器需要保存组播树信息,参加组播树的维护过程.组播树上的其他路由器只是路由组播数据包,不维护组播树信息和参与其维护,因此,ODBM 节省了网络资源.此外,它与现有的组播路由相比还有如下优点:对网络拓扑变化的适应性,若非关键节点失效,则由 Internet 路由自行解决传输路径的相应变化,不需要像现有组播树维护方式那样为适应这种变化要作大量维护工作;扩展性好.可以用于 Internet 范围组播.中间节点利用现有的单播路由传输组播包,就把原来基于平面的网络结构维护方式转变为层

次型结构,从而有很好的扩展性;ODBM 与现有的组播模型具有很好的互操作性[15].它可以与现有组播模型共存,利用组播地址目的选项来实现算法使用的透明性,可以与现有组播模型互相转换,还可以通过适当的修改用于各种类型的组播树,例如共享树和有源树.

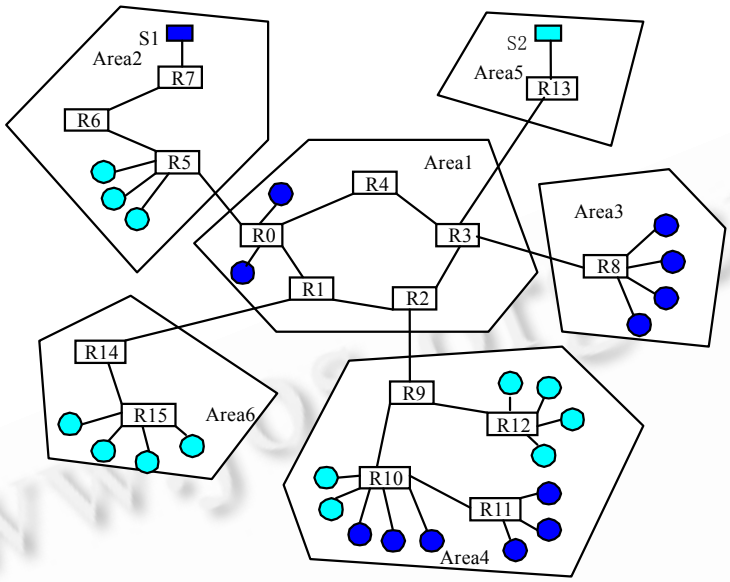

Fig.6 Network topological structure

图 6 网络拓扑结构

目前,组播路由中尚有许多问题没有得到完满的解决,在 Internet 所有成员都能使用 IP 组播之前仍有许多工作要做,例如,域间组播路由的技术问题需要解决,组播中的可靠性、安全性等问题也需要进一步完善和解决.因此,提供更加完善、有效的组播解决方案,是因特网发展的需要.

References:

[1] Mohammad Banikazemi. IP multicasting: concepts, algorithms, and protocols; IGMP, PRM, CBT, DVMRP, MOSPF, PIM, MBONE. 2000. http://www.cis.ohio-state.edu/~jain/cis788-97/ip_multicast/index.htm.

[2] Sahasrabuddhe LH, Mukherjee B. Multicast routing algorithms and protocols: a tutorial. IEEE Network, 2000,14 (1):90~102.

[3] Moy J. Multicast extensions to OSPF. RFC 1584, 1994.

[4] Waitzman D, Partridge C, Deering SE. Distance vector multicast routing protocol. FRC 1075, 1988.

[5] Williamson B. Developing IP Multicast Networks. Indianapolis, IN: Cisco Press, 1999.

[6] Almeroth KC. The evolution of multicast: from the MBone to interdomain multicast to Internet2 deployment. IEEE Network, 2000, 14(1):10~20.

[7] Diot C, Levine BN, Lyles B, Kassem H, Balensiefen D. Deployment issues for the IP multicast service and architecture. IEEE Network, 2000,14(1):78~88.

[8] Adams A, Siadak W, Nicholas J. Protocol independent multicast——dense mode (PIM-DM) protocol specification (Revised). Internet-Draft, 2002. http://www.rfc.editor.org/cgi-bin/iddoctype.p1?letsgo=draft-ietf-pim-dm-new-v2-02.

[9] Handley M, Fenner B, Kouvelas I, Holbrook H. Protocol independent multicast-sparse mode PIM-SM: protocol specification (Revised). Internet-Draft, 2002. http://www.rfc.editor.org/cgi-bin/iddoctype.p1?letsgo=draft-ietf-pim-sm-new-v2-06.

[10] Ballardie A. Core cased trees (CBT version 2) multicast routing. RFC 2189, 1997.

[11] Ballardie A. Core based trees (CBT) multicast routing architecture. RFC 2201, 1997.

[12] Meyer R. PARSEC user manual. 1998. http://pcl.cs.ucla.edu/projects/parsec/manual.

[13] http://pcl.cs.ucla.edu/projects/parsec/. 2001.

[14] Estrin D, Farinacci D, Helmy A, Thaler D, Deering S, Handley M, Jacobson V, Liu G, Sharma P, Wei L. Protocol independent multicast-sparse mode (PIM-SM): protocol specification. RFC 2362, 1998.

[15] Aggarwal S, Paul S, Massey D, Caldararu D. A flexible multicast routing protocol for group communication. Computer Networks, 2000,32(1):35~60.