ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.8, August 2010, pp.2010−2022
doi: 10.3724/SP.J.1001.2010.03594

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种拓扑感知的应用层组播方案*

张新常[1,2+], 王 正[1,2], 罗万明[1,3], 阎保平[1]

[1](中国科学院 计算机网络信息中心,北京 100190)

[2](中国科学院 研究生院,北京 100049)

[3](中国互联网络信息中心,北京 100190)

Topology-Aware Application Layer Multicast Scheme

ZHANG Xin-Chang[1,2+], WANG Zheng[1,2], LUO Wan-Ming[1,3], YAN Bao-Ping[1]

[1](Computer Network Information Center, The Chinese Academy of Sciences, Beijing 100190, China)

[2](Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)

[3](China Internet Network Information Center, Beijing 100190, China)

+ Corresponding author: E-mail: xinczhang@hotmail.com

Zhang XC, Wang Z, Luo WM, Yan BP. Topology-Aware application layer multicast scheme. *Journal of Software*, 2010,21(8):2010−2022. http://www.jos.org.cn/1000-9825/3594.htm

Abstract: This paper proposes a topology-aware clustering model (called TCM). Furthermore, it proposes a TCM-based application layer multicast scheme (called TCMM). In TCMM, many nearby nodes are clustered, which localizes the transport of some nodes and alleviates the negative impact caused by different join sequences. Analysis and experiments show that TCMM can effectively group nearby nodes and build multicast trees with similar gross performance in different join orders. In addition, TCMM can improve some of other multicast performance in some degree.

Key words: application layer multicast; delivery tree; clustering; network topology; overlay network

摘 要: 提出了一种具有拓扑感知能力的拓扑簇模型 TCM(topology-aware clustering model),并在此基础上提出了一种有效的应用层组播方案 TCMM(TCM-based multicast).TCMM 能够将一些相近的节点组织在一个拓扑簇中,从而在一定程度上实现了数据包的本地传输,并能缓解不同加入顺序对转发树的不利影响.分析和实验结果表明,TCMM 能够实现有效的聚簇,能够在不同的加入顺序下构造性能大体一致的转发树,并能在不同程度上改善其他一些组播性能指标.

关键词: 应用层组播;转发树;聚簇;网络拓扑;覆盖网络

中图法分类号: TP393 文献标识码: A

从组播效率角度来看,IP 组播是实现 Internet 范围内组通信的最佳方式.然而,出于多种原因 (如计费困难、过渡消耗路由器资源等),ISPs 往往限制组播路由功能,从而限制了 IP 组播在 Internet 上的广泛部署.IP 组播的更多缺陷参见文献[1].近年来,出现了 IP 组播的替代方案——应用层组播.应用层组播无须对路由器作任何修改,

* Supported by the National Basic Research Program of China under Grant No.2003CB314807 (国家重点基础研究发展计划(973))
Received 2008-08-09; Revised 2008-10-24; Accepted 2009-02-24

因此在 Internet 上非常容易部署.在应用层组播中,一个主机需要向其子节点(主机)发送数据包,且其子节点数量是有限制的.在有度限定的前提下,构建最小延迟(或树代价)的组播转发树是一个 NP 难问题[2−4].此外,应用层组播还面临如下的实际问题:(1) 由于主机不知道底层网络的拓扑信息,相近的节点可能分布在组播转发树中相距较远的位置,从而造成组播性能的下降;(2) 群组成员的加入是一个渐进过程,新加入者在很大程度上依靠已存在节点的信息来确定在组播转发树中的位置,即不同的加入顺序直接影响组播转发树的结构和性能.本文提出了一种具有拓扑感知能力的聚簇模型,通过这种拓扑启发式来聚合相近的节点,并缓解已有节点对新加入者的不良影响.基于这种聚簇模型,本文进一步提出了一种有效的应用层组播方案,对其进行了一定程度的性能分析,并通过实验验证了方案的有效性.

1 相关工作

由于应用层组播能够很容易地在 Internet 上部署,近年来得到了广泛的研究,出现了较多的应用层组播协议或解决方案,如 HMTP[5],NICE[6],NARADA[7],TBCP[8],SCRIBE[9],ZIGZAG[10],OMNI[11],TOMA[12],HOMP[13],PALM[14].文献[15]指出了应用层组播重点优化的性能指标,包括优化树代价、延迟、Stress 等.一些协议可能同时优化多个指标,如 NICE 同时优化延迟和 Stress,而 HMTP 优化树代价和延迟.

NICE 和 ZIGZAG 协议显式地利用分层聚簇的思想.NICE 协议将形成的覆盖网络组织成分层的簇结构,并根据该层次形成转发树.在 NICE 中,每个簇所包含的成员数量为[k,3k−1].如果一个簇的尺寸超出该范围,则执行相应的合并或分裂过程.每个 NICE 簇都有一个簇头节点,负责转发不同簇间的数据包.在 NICE 簇的分层结构中,某层次的簇是由下一层次簇的簇头所形成的.在各层次的簇中,需要较高的代价来维护相应的成员关系.ZIGZAG 协议的簇层次构造与 NICE 类似,每个簇尺寸都有限制,并服从自底向上构造层次的原则.当新成员加入群组时,上述两种协议均先用深度优先搜索法将新成员定位到最底层次的簇.如果加入最底层次的簇时破坏了簇尺寸的范围限制,则进行相应的分裂或合并操作.由上述聚簇过程可以看出,NICE 和 ZIGZAG 协议中的簇结构具有以下主要缺陷:

(1) 在某些情况下不能聚合相近的节点.如果在某局部范围内有若干成员节点,其数量超过簇尺寸的限制,则需要多个簇来覆盖这一范围,并且很难保证这些簇中不包含相距较远的成员节点.此外,由于在一个组播应用中,成员具有较高的动态性,不断的簇分裂和合并操作将加剧上述不能聚合相近节点的情况.

(2) 簇需要较高的维护代价.由于簇分裂和合并操作都需要考虑重新选择簇的中心(簇头),簇内的成员需要保持紧密的联系.同时,当底层的簇产生变化时,高层的簇可能会随之变化.因此,簇的维护代价较高.

HMTP 协议是为多到多点组播应用设计的,但其同时包含了点到多点的应用场景.HMTP 采用一种递归的贪婪 DFS(depth first searching)来定位新加入者的位置:新加入者测量到候选父节点及其子节点的距离(初始时候选父节点为树根),如果到候选父节点的距离在所测距离中最近,则向其发送加入请求;若当前候选父节点不接受加入请求,则选择最近的子节点作为下一个候选父节点,然后重复上述过程.贪婪 DFS 方法能使新加入者在加入过程中获得较好的位置,从而形成较好的转发树结构.此外,HMTP 对成员的加入顺序作了一定的考虑,并通过附加的改进过程来缓解不同加入顺序所带来的负面影响.由于在改进过程中随机选择下一候选父节点,HMTP 需要较长时间才能达到稳定的状态.因此,HMTP 对不同加入顺序的适应能力并不强.与 HMTP 类似,TBCP 的加入过程也是一个递归过程,通过一种评估机制确定成员在组播转发树中的位置.

TOMA 提出了一种两层的 overlay 组播体系.第 1 个层次是组播服务覆盖网络 MSON(multicast service overlay network),它由若干代表服务域的代理所组成.当一个主机想加入群组时,首先选择一个服务域,然后在该域内通过应用层组播协议形成第 2 层结构(数据转发树).OMNI 组播方案与 TOMA 类似,都是通过在 Internet 上部署代理来提高组播的应用规模和转发性能.

2 拓扑簇模型及相应的概念

在应用层组播方案中,聚簇是一种良好的解决思路.由上文对相关工作的介绍可以看出,目前已有的聚簇方

案大体上可以分为两类:逻辑聚簇(如 NICE 和 ZIGZAG)和代理聚簇(如 OMNI 和 TOMA).由于各代理将组播范围划成若干拓扑感知(topology-aware)的区域,后一种聚簇方案组播效果相对较好.然而代理聚簇方案依赖于特殊设施,使得其应用范围有一定的限制.本文提出了一种新的聚簇模型——拓扑簇模型,以聚合在拓扑意义上相近的节点.

拓扑簇模型具有两个层次的簇结构,即拓扑簇层次和群组层次,其中,群组层次由所有拓扑簇的头(cluster leader)节点组成,拓扑簇层次指各个具体的拓扑簇.因此,拓扑簇模型的层次结构与代理聚簇的层次结构类似.与代理聚簇不同,拓扑簇是自动形成的,而不是由代理限定.一个拓扑簇覆盖一定范围的网络区域,故其能够聚合相邻节点.

图 1 显示了拓扑簇模型的基本结构.该图包含 11 个(A~K)个拓扑簇,各簇有且仅有 1 个头节点.除了头节点,一个拓扑簇还可能包含 1 个或多个传输代理 TA(transport agent)和一般簇成员 CM(common member).如簇 B 包含 1 个头节点(节点 1)、1 个传输代理(节点 2)和 3 个一般成员(节点 3~节点 5).拓扑簇间存在一些相关连接,形成了组播转发树和该转发树的维护结构.如在图 1 中,所有节点和连接不同节点的实线组成了一棵组播转发树,所有节点和不同节点的连线(包括实线和虚线)组成了对应转发树的维护结构.下一节将对拓扑簇模型进行详细的介绍,拓扑簇的具体形成过程也将在下一节说明.

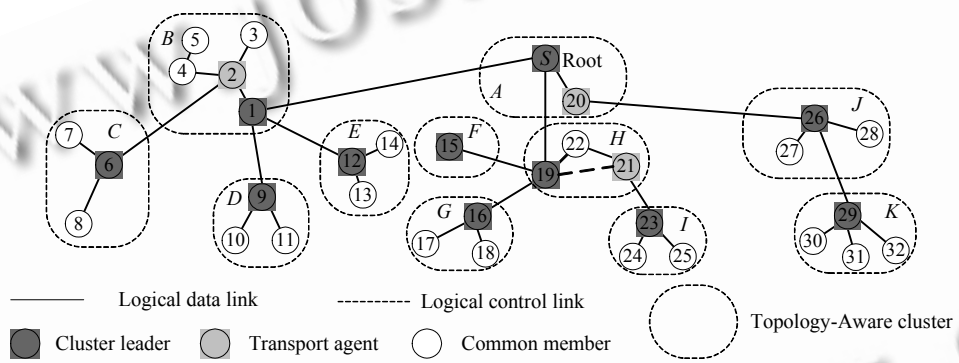

Fig.1 Topology-Aware clustering model for application layer multicast

图 1 应用层组播中的拓扑簇模型

定义 **1**(拓扑簇). 拓扑簇是由相邻的节点组成的集合.在一个拓扑簇中有 3 类节点:头节点、传输代理和一般成员节点,且满足:(1) 一个拓扑簇有且仅有 1 个头节点,任意头节点仅属于 1 个拓扑簇;(2) 一个拓扑簇可以包含 $n(n\geq 0)$个传输代理或一般成员,且任意传输代理或一般成员仅属于 1 个拓扑簇;(3) 设拓扑簇的头节点为 h,则对任意 $m(m\in S(h))$,$d(h,m)\leq\lambda$.

在定义 1 中,$S(h)$表示以 h 为头节点的拓扑簇中所有传输代理和一般成员的集合,λ 表示拓扑簇的范围阈值,$d(h,m)$从节点 h 到 m 的单播距离.λ的取值没有严格的规定,但其限制范围应尽可能地覆盖中等规模的网络.在拓扑簇中,TA 和 CM 并不是必须的.如在图 1 中,拓扑簇 A 没有 CM 节点,拓扑簇 F 仅有 leader 节点.

定义 **2**(**CM**). CM 是一个拓扑簇的非头节点,且在组播转发树中,其父节点属于同一拓扑簇.此外,如果 CM 有子节点,则这些子节点与该 CM 属于同一拓扑簇.如在图 1 中,节点 4 和节点 27 均为 CM 节点.

定义 **3**(**TA**). TA 是一个拓扑簇的非头节点,并且它在组播转发树中至少包含 1 个非同拓扑簇的子节点.

在本文中,当且仅当某节点从其另一节点接收数据包时,称前者是后者的子节点.图 1 给出了 3 个 TA 节点(节点 2、节点 20 和节点 21),其各自有一个非同簇子节点.注意:(1) TA 节点是其所在拓扑簇 leader 的传输代理节点;(2) TA 节点可能同时为其所在拓扑簇 leader 的传输代理节点和子节点.当一个成员想成为某 leader 的非同簇子节点时,如果该 leader 不能接受更多的子节点,则考虑让其传输代理接受上述成员的加入请求.换句话说,代理节点起到替其所属簇的 leader 接受更多非同簇子节点的作用.

定义 **4**(**leader**). leader 是对应拓扑簇的中心节点.如果某 leader 不是组播源,则它从某个其他簇的 leader 接收数据包,并向其子节点转发收到的数据包.同时,leader 记录其拓扑簇内的 TAs 地址.

在本文中,如果两个群组成员存在直接的关联,则称这两个节点间的单播路径为逻辑链路.逻辑链路包含两种类型,即逻辑数据链路和逻辑控制链路.

定义 **5**(逻辑数据链路). 假定存在两个成员节点 A 和 B,如果一个节点 A 向另一个节点 B 转发数据包,则称两个节点间的逻辑链路为逻辑数据链路.其中,分别称 A 和 B 为对应逻辑数据链路的起点和终点.

定义 **6**(逻辑控制链路). 在一个拓扑簇中,leader 与 Tas 之间的逻辑链路为逻辑控制链路.

根据定义 5 和定义 6 可知,组播转发树由所有逻辑数据链路组成,数据包将按照该树进行转发.而逻辑控制链路不转发组播数据包,它仅在数据转发树的维护和构建过程中发挥作用.

定义 **7**(簇点树). 如果把拓扑簇看作簇点,则转发树可以认为是由所有簇点所组成的树,称为簇点树.如可以把如图 1 所示的转发树看作是由 11 个簇点和连接不同簇之间的逻辑数据链路所组成的树.

我们用记号 $C_e(h)$表示由所有 h 的非同簇子节点组成的集合,$C_c(h)$表示由所有 h 的同簇子节点组成的集合,$C_a(h)$表示由所有同时为 h 的子节点和传输代理的节点所组成的集合.

定理 **1**. 拓扑簇数量随着群组规模变大而增长,但其增长逐渐趋缓,直至达到零增长.

证明:假定Ω为组播可达的拓扑空间,显然Ω是一个有限空间.令 N_i 表示第 i 个加入的成员,C_i 表示前 i 个成员加入后已有拓扑簇所覆盖的空间.根据 TCM 模型,可知 $C_{i+1}=C_i+\Delta(\Delta\geq 0)$.当 N_{i+1} 不能加入到已有拓扑簇时,易知 $\Delta>0$,$\Omega-C_{i+1}<\Omega-C_i$.因此,当加入的成员在$\Omega$内足够分散且达到一定规模时,已有拓扑簇所覆盖的空间与Ω重叠,拓扑簇数量不再增长.另外,由于$\Omega-C_j$ 值随着 j 不断增大而非递增下降,新成员新建拓扑簇的概率也随着群组规模的增加而非递增下降,即拓扑簇数量增长逐渐趋缓. □

定理 1 给出了拓扑簇模型的一般性质.在不同的网络描述模型下,拓扑簇数量收敛速度有所不同.为了准确地描述现实网络,网络模型至少是三维的,进一步的内容可参见文献[16].在本文相关的证明或描述中,我们不限定某一种网络模型.

3 基于 TCM 模型的组播方案

本文提出的基于 TCM 模型的组播方案(TCM-based multicast,简称 TCMM)将重点讨论组播转发树的构建与维护.引入 TCM 模型后,TCMM 将在两个层次进行转发树构建:在簇内构建转发树(称为簇内树);在簇间构建相应的簇点树.上述两类树的并集形成了组播转发树.从本质上讲,两类树都可以采用多数现有的组播协议来实现.由于 HMTP 协议在定位过程中具有一定的拓扑感知能力,并具有较低的维护代价,本文提出的基于 TCM 模型的组播方案(TCMM)以 HMTP 为原型来实现簇内树和簇点树的构造.在组播应用中,转发树的建立是一个动态过程,即随着成员的不断加入而建立相应的转发树.因此,转发树构建实际上是一个不断响应成员加入的过程.

3.1 加入请求的处理

当一个新的节点 N 想成为某个已存在节点 E 的子节点时,它向 E 发送加入请求消息 $joinreq(E,N,T)$.其中,T 表示想成为 E 的子节点类型,值为 1 表示非同簇子节点,值为 0 表示同簇子节点.E 按照以下规则处理节点加入请求:

规则 **1**. 如果 E 的剩余扇出度大于 0,则在收到 $joinreq(E,N,T)$消息后,无论 T 为何值,E 都接受 N 的加入请求,同时向 N 返回加入成功消息.

在规则 1 中,一个节点的扇出度是指其最多能接纳子节点的数量(参见文献[5]),而剩余扇出度则指其扇出度减去已有子节点数量的值.容易看出,规则 1 充分利用了成员的剩余扇出度.

规则 **2**. 假定 E 的剩余扇出度为 0,且收到加入消息 $joinreq(E,N,0)$,则按照以下情况处理:(1) 若$|C_c(E)|>0$,则 E 向 N 返回加入失败消息;(2) 若$|C_c(E)|=0$,则 E 向随机指定的一个非同簇子节点发送 *redirect* 消息,并从子节点列表中删除该节点,然后接受 N 的加入请求,向 N 返回加入成功消息.

当一个节点收到 *redirect* 消息后,从其父节点开始重新加入群组.为了在新的父节点没有找到之前不中断数据包的接收,可以在让触发 *redirect* 消息发送的新加入者临时向其发送数据包.此外,当一个成员收到加入成功消息时,记录其节点类型.本文不再描述具体细节.

规则 **3**. 假定 E 是 TA 节点,且收到加入消息 *joinreq*(E,N,1),则 E 处理如下:(1) 如果 E 的剩余扇出度为 0,且$|C_c(E)|\leq 1$,则 E 向 N 返回加入失败消息;(2) 如果 E 的剩余扇出度为 0,且 $f_e(E)>0$,则 E 向随机指定的一个同簇但不是 TA 的子节点发送 *redirect* 消息,并从子节点列表中删除该节点,然后接受 N 的加入请求,向 N 返回加入成功消息.

在本文中,函数 $f(n)$表示节点 n 的扇出度,函数 $f_e(n)$表示节点 n 接纳非同簇子节点的能力.具体定义为

$$f_e(n)=\begin{cases} f(n)-|C_e(n)|-|C_a(n)|, & \text{if } n\text{有TA子节点} \\ f(n)-|C_e(n)|-1, & \text{否则} \end{cases},$$

其中,节点 n 的一个 TA 子节点 c 满足:c 为 n 所在簇头节点的传输代理;c 为 n 的一个子节点.

规则 **4**. 假定 E 是 leader 节点,且收到加入消息 *joinreq*(E,N,1),则按照以下情况处理:(1) 如果 E 的剩余扇出度为 0,且$|C_c(E)|\leq 1$,则:(a) 如果 E 已经有传输代理 TA,则向 N 返回 *graft* 消息,该消息中包含 TA 节点地址;(b) 如果 E 没有传输代理 TA,且有扇出度大于 1 的同簇子节点,则指定一个扇出度大于 1 的同簇子节点为 TA 节点,随后发送 *graft* 消息.否则,发送加入失败消息.(2) 如果 E 的剩余扇出度为 0,且 $f_e(E)>0$,则 E 向随机指定的一个同簇但不是 TA 的子节点发送 *redirect* 消息,并从子节点列表中删除该节点,然后接受 N 的加入请求,向 N 返回加入成功消息.(3) 在其他情况下,E 向 N 返回加入失败消息.

注意,只有 leaders 和 TAs 能收到 *joinreq*(E,N,1),且 *graft* 消息使新加入者可能选择合适的 TA 作为候选父节点,以便继续加入过程,具体见下文的加入算法.另外,从上述规则可以看出,非叶子 leaders 和 TAs 节点的扇出度至少为 2.为了保证该需求和提高群组组播的效率,扇出度小于 2 的 leader 将被推向转发树的底部(称为下推原则),本文将不再对其作更多的介绍.

规则 2~规则 4 保证了 leader 节点在其所在簇有其他成员时,具备至少接受 1 个同簇子节点的能力,从而实现向该簇中非头节点转发组播数据包.令 $N(n,\Gamma)$表示在空间Γ内可以用节点 n 作为其上游节点的节点数量,$E(N(n,\Gamma))$表示 $N(n,\Gamma)$的数学期望值,Ω为组播可达的拓扑空间,可得定理 2.

定理 **2**. 若成员在Ω内均匀分布,则 $E(N(e,\Omega))>E(N(i,\Omega))$,其中,$e$ 为任意 leader 节点,i 是任意非 leader 节点.

证明:由拓扑簇模型可知,各拓扑簇空间大小相等.已知成员在Ω内均匀分布,故各拓扑簇空间内所含非 leader 成员数量的数学期望相同,设为μ.进一步可知,在任意拓扑簇空间 T 内,若 n 为 leader,则 $E(N(n,T)=\mu$;若 m 为非 leader 节点,则 $E(N(m,T))\leq\max\{0,\mu-1\}$.假设 e 在拓扑簇空间 T_1 中,i 在簇空间 T_2 中,则 $E(N(e,\Omega))>E(N(e,T_1))=\mu$,$E(N(i,\Omega))=E(N(i,T_2))\leq\max\{0,\mu-1\}$.因此,$E(N(e,\Omega))>E(N(i,\Omega))$成立. □

根据定理 2 所述的启发式,规则 3 和规则 4 保证 leader 和 TA 节点具有优先接纳非同簇子节点的能力,按照贪心策略来降低后来成员加入群组后的节点平均延迟.同理可知,在一个拓扑簇空间 T_i 内,在同样的条件下,TA 节点 a 的 $E(N(a,\Omega))$大于 CM 节点 b 的 $E(N(b,\Omega))$.因此,规则 3 和规则 4 选择 CM 节点作为被替代节点.此外,保留 TA 节点还可以尽可能地减少替换操作所影响的节点数量.

在本文中,我们用簇点对外扇出度表示以该簇内节点(包括 leader 和 TA 两种节点)为父节点的最大非同簇子节点(其他簇的 leader 节点)数量.

定理 **3**. 如果一个簇 C 仅包含 1 个 leader 或包含至少 1 个扇出度大于 1 的非簇头节点,则该簇点对外扇出度不比 leader 的扇出度低.

证明:设簇 C 的 leader 节点为 L,其扇出度为 f.当簇 C 包含至少 1 个扇出度大于 1 的非簇头节点时,根据上文所述下推原则,L 至少有 1 个扇出度大于 1 的同簇子节点,即存在潜在的 TA 节点.根据规则 4,L 的最大对外扇出度=f–1.而 TA 节点具备接受至少 1 个非同簇节点的能力.因此,在这种情况下,该簇点的对外扇出度不比 leader 的扇出度低.当簇 C 仅包含 1 个 leader 时,簇点对外扇出度等于该簇 leader 的扇出度.综上所述,该定理成立. □

定理 3 给出了引入 TA 节点的原因.节点的扇出度在不同组播协议中会产生不同的影响,然而对 HMTP 协

议而言,较高的扇出度能够提高其组播效率(参见文献[15]).扇出度之所以对 HMTP 产生正面的影响,最主要的原因是其贪心 DFS 方法具有一定的拓扑感知能力.

3.2 成员加入

如果一个新加入者 N 想加入一个群组,它首先联系一个 RP(rendezvous point)以获得目前群组的 *ROOT* 地址,然后按照 *Join*(*ROOT*,*N*)过程加入群组.

Procedure: *Join*(*ROOT*,*N*)

1. 初始化:初始化一个工作栈 S,$p\leftarrow ROOT$. //S 用于存放候选父节点,p 表示当前候选父节点.
2. while (p is not null)
3. 测量从节点 p 到节点 N 的距离 $d(p,N)$.
4. if ($d(p,N)\leq\lambda$)
5. 按照 *Joincluster*(*p*,*N*)加入以 p 为头节点的拓扑簇.
6. else
7. 节点 N 查询并得到节点 p 的非同簇子节点列表 L.
8. 如果 L 不为空,则测量从 L 中各节点到节点 N 的距离.
9. 设 N 到 L(不为空)中节点和 p 的最小距离为 $d_{\min}$,如果 L 为空或 $d(p,N)=d_{\min}$,则 N 向 p 发送 *joinreq*(*p*,*N*,1);若 p 返回加入成功消息,则结束加入过程.如果返回 TA 节点(收到上文所述 *graft* 消息),则测量从各 TA 节点到 N 的距离.
10. 将 L 中的子节点和 TA 节点(如果有)按照距离降序压入栈 S 中.
11. $p\leftarrow top(S)$ //取出栈顶节点
12. end if
13. end whild

在上述过程中,第 10 步的执行条件包含两种情况:第 9 步中收到 p 发送的加入失败消息;第 9 步中收到 p 发送的 *graft* 消息.此外,*Join*(*ROOT*,*N*)过程包含簇加入子过程 *Joincluster*(*p*,*N*).该子过程类似于 *Join*(*p*,*N*),但是以下几点不同:语句 7 得到 p 的同簇子节点 L;语句 4 的判断条件用 false 替代;在语句 9 中用 *joinreq*(.,.,0)代替 *joinreq*(.,.,1);删除与 TA 节点有关的内容.对于 *joinreq* 消息的处理,我们已在第 3.1 节中加以介绍.

由加入过程可知,拓扑簇在两个层次上对新加入者的定位产生良性影响:(1) 一旦确定新加入者从属于某拓扑簇,则后继的定位将限制在该簇内;(2) 在簇点树级别上,由于簇点数量小于成员数量,已有成员对新加入者定位的不利影响被削弱.此外还可以看出,拓扑簇的建立是在加入过程中随着一个成员成为 leader 自动完成的.上述过程省略了簇头选择过程,从而降低了簇的维护负担.实际上,拓扑簇也不需要分裂和合并过程(见定理 4),从而进一步降低了簇维护代价.

定理 **4**. 在加入过程中,拓扑簇的大小不受簇内成员数量的限制,而是取决于该簇所覆盖的拓扑范围.

证明:根据加入过程的第 9 步,当一个节点 N 向 p 发送 *joinreq*(*p*,*N*,1)并收到加入成功消息后,该节点成为一个 leader 节点,从而建立所在的拓扑簇.假设存在某新加入成员 M,满足 $d(N,M)\leq\lambda$(注意,簇范围阈值λ已知).只要 M 选择 N 作为候选父节点,则根据加入过程第 4 步和第 5 步,无论以 N 为头的拓扑簇内有多少已有成员,M 都成为该簇的 TA 或 CM 节点.因此,拓扑簇的大小取决于由 leader 和λ所确定的簇覆盖范围. □

若在加入过程中,转发树中 leader 节点数量不再变化,与 leader 相关的树边也保持不变,则称该群组的簇点树已经达到稳定状态.

定理 **5**. 若簇点树达到稳定状态,则此后新成员节点无论以何种顺序按照 TCMM 加入群组,其所属的拓扑簇都是固定的.

证明:按照 TCMM,成员加入时首先试图找到合适的 leader 以便加入该 leader 所在的簇;如果没有找到这样的 leader,则该成员充当一个新的 leader.在簇点树稳定的情况下,所有 leader 及其位置已经确定,故新成员加入实际上是发现一个合适的 leader 并加入其所在簇的过程.按照 TCMM,查找合适 leader 的过程是在簇点树上按照

同一套协议进行的,从而其找到的所属拓扑簇是固定的. □

定理 4 和定理 5 说明了本文提出的应用层组播方案对加入顺序具有较好的适应能力,并且同一群组所生成的组播转发树都具有较好的拓扑感知(topology-awareness)能力.

3.3 结构改进

在本文提出的方案中,采用附加的重新加入过程来改善初始的转发树.重新加入过程由成员节点定期执行.该过程与加入过程基本类似,但在以下几个方面有所不同:

(1) 从 root path(从根到某节点的路径,见文献[5])中随机找到一个节点开始重新加入过程,以便给 *ROOT* 及其邻近节点减轻压力.

(2) 设非簇头节点 N 新找到的父节点为 p',当前的父节点为 p.当满足 $d(p',N)\leq d(p,N)-\sigma$ 时,切换到新的父节点.其中,σ 为收益阈值,其值大于 0.

(3) 如果一个 leader 在重新加入过程中发现能够加入到另一个簇 C 中,则加入簇 C,同时成为簇 C 头节点的一个传输代理.

(4) 如果 leader 节点或 TA 节点 i 切换到新的拓扑簇 C,则在其当前簇中该节点以下的 TA 节点和该节点都成为其新所属簇的 TA 节点.此外,为保持 C 的原有结构,i 不选择含有 TA 子节点的节点作为其父节点.

(5) 在向工作栈压入新的候选节点时(加入过程语句 11),不是按照一定的顺序,而是采用随机的顺序,以便找到潜在的更好的父节点.

当一个带有同簇子节点的节点切换到新的非同簇父节点时,其同簇子节点也被动地加入到新的簇,从而可能加大新簇的簇尺寸.为了限制这种簇大小的偏移,TA 节点和 CM 节点定期地向其簇的头节点发送测量消息,以便核对到头节点的当前距离是否还满足上文的簇范围阈值.如果超过该值,则从其簇的头节点开始执行 *Join* 过程.

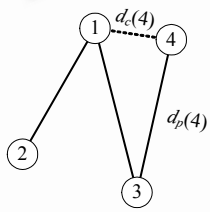

Fig.2 An example of V shape

图 2 V 形的实例

多数协议通过不断探查群组中已有的成员来确定自己的位置,因此已存在的成员对新加入者的影响较大.尽管存在很近的节点,一个新加入者也可能被定位到离其很远的已存在节点,从而形成“V”形,如图 2 所示.更加形式化地,假定一个节点 n 到其父节点的距离是 $d_p(n)$,其最近的已存在节点的距离为 $d_c(n)$,如果 $d_p(n)/d_c(n)\geq\delta$(δ为大于 1 的常数),则称该节点 n 对应一个“V”形,记为 V(n).根据 TCM 模型和本文对转发树的构造及改进算法,可以得到如下定理:

定理 6. 在稳定的 TCMM 转发树中不存在同时满足 $d_c(n)\leq\lambda$ 和 $n\in L$ 条件的 V(n)形.其中,L 为包含所有 leader 节点的集合.

证明:假设在稳定的 TCMM 转发树中存在某 leader 节点 n,满足 $d_p(n)/d_c(n)\geq\delta$,且 $d_c(n)\leq\lambda$.设离 n 最近的节点为 k.根据上文所述,n 会周期性地执行重新加入过程,一旦在该过程中选择 k 作为候选父节点,则 n 加入以 k 为簇头的拓扑簇.因此上述假设不成立,即该定理得证. □

另外,在一个拓扑簇内,由于成员数量相当较少,并且涉及的拓扑复杂度相对于整个群组而言比较简单,“V”形的数量及相应 d_p/d_c 比值均会下降.

3.4 结构维护

在某节点的子节点列表中,其各个子节点表项都对应一个定时器.每个节点 n 定期地向其父节点 p 发送 *live*(p,n)消息以宣称自己的存活.当收到该消息后,p 更新对应节点的定时器.如果某子节点的定时器超过预定的阈值,则删除该子节点.类似地,TA 节点也需要定期地向相应簇头节点发送 *live* 消息.

当某节点 n 退出群组时,应该向其父节点和子节点发送 *leave* 消息通告该事件.当父节点收到该消息后,将节点 n 的信息删除.当子节点收到该消息后,从其 root path 中找到最近的、存活的节点,然后从该节点开始执行加入过程以便加入到该群组.如果一个节点是 TA 节点,则除了向上述节点发送 *leave* 消息以外,还同时向其簇的头

节点发送该消息,以便后者进行相应的处理.

然而,节点 n 在退出时可能由于某些原因没有发送 *leave* 消息,则上述操作需要主动地完成.如上文所述,当子节点或 TA 子节点在一段时间内没有发送 *live* 时,则删除其相应信息,由此主动删除了未通告离开的子节点.由于非 *ROOT* 节点需要从其父节点接收数据包,故可籍此诊断其父节点的存活.一旦某节点断定其父节点已经未加通知地离开,则主动加入到新的父节点.

其他维护操作如 root path 的更新等,本文提出的方案将采用类似 HMTP 的方法,这里不再赘述.

4 实验结果分析

在本部分模拟实验中,我们用 GT-ITM[17]生成了 4 个分别包含 4 800 个节点的拓扑结构.针对上述各个拓扑结构,分别生成 1 000 个节点(代表主机节点),各主机节点按照一定的策略与对应拓扑中的 stub-domain 节点相连接,形成 4 个包含 1 000 个主机和 4 800 个路由器节点的拓扑结构,并用拓扑 1、拓扑 2、拓扑 3 和拓扑 4 加以标识.在拓扑 1、拓扑 2、拓扑 3 和拓扑 4 中,节点的平均度分别为 3.9,4,3.8 和 4.在拓扑 1、拓扑 2 和拓扑 3 中,主机节点随机连接一个 stub-domain 节点,并且保证一个 stub-domain 节点最多连接 1 个主机节点.在拓扑 4 中,随机选择 800 个 stub 节点,然后主机节点在其中随机选择连接的节点,并保证各 stub-domain 节点至少连接 1 个主机节点.在本文模拟实验中,我们用延迟作为距离度量.我们用 NS-2[18]实现对相关协议的模拟,并对其实验结果进行分析.此外,我们用取自 PlanetLab 测试床的 ping 数据集[19]对 TCMM 的聚簇特性进行验证.

4.1 聚簇效果

4.1.1 模拟实验

为了定量地分析 TCMM 聚簇效果,引入聚簇失败率 CFR(cluster failure ratio)、聚簇权值 CW(clustering weight)和聚簇偏移量 CO(clustering offset)概念.

定义 **8**(聚簇失败率). 转发树的聚簇失败率 CFR=应该但没有被聚簇的节点数量/应该被聚簇的节点数量.其中,一个应该但没有被聚簇的节点 n 满足:n 没有和其他节点组成一个拓扑簇,且$\exists(m)(d(m,n)\leq\lambda,m$ 是头节点)成立.类似地,一个应该聚簇的节点是指能够与其他节点聚簇的节点,即能按照 TCM 模型定位在一个已有拓扑簇中.注意,在 TCMM 中,如果一个节点不能和其他节点聚簇,则其是一个簇(仅包含 1 个节点)的头节点.

定义 **9**(聚簇权值). 任意节点 n 的聚簇权值 $CW(n)$定义为

$$CW(n)=\begin{cases}w(n), & \text{如果}n\text{被聚簇,且不是头节点}\\ 0, & \text{其他}\end{cases},$$

其中,$w(n)=(\lambda/d(h,n))^{\frac{1}{2}}$,$h(h\neq n)$是 n 所在簇的头节点.类似地,$CW'(n)$表示 n 在被最佳聚簇情况下的聚簇权值.所谓 n 被最佳聚簇是指没有其他任何头节点比当前头节点到 n 的距离最短.

定义 **10**(聚簇偏移量). 转发树的聚簇偏移量 CO 定义为

$$CO=\sum_{n\in U}(CW'(n)-CW(n)),$$

其中,U 为应该但未被聚簇的节点和被聚簇但不是被最佳聚簇的节点的集合.

在本实验中,采用的簇范围阈值λ为 1 400ms,该值低于不同 transit 域之间的链路延迟,同时低于部分不同 stub 域之间的链路.在组播应用中,由于群组成员具有较高的动态性,初始构造的转发树质量非常重要.因此,尽管改进过程能够提高组播性能,本文中的实验数据也仅针对初始构造的转发树.

图 3 显示了 TCMM 在 10 个不同群组规模、不同拓扑结构下的聚簇失败情况,图 4 则显示了对应的被聚簇的节点数量.由图 3 和图 4 可以看出,TCMM 能够实现较好的聚簇.在规模较小的群组下,CFR 可以达到 0,表示没有任何应该聚簇的节点被疏忽.随着群组规模的增大,CFR 值会不同程度地增长,但总体上仍保持较低的值.另一方面,虽然在规模较大的群组下 CFR 值相对较大,但其被聚簇节点数量也迅速增加.

图 5 从另一个角度反映了 TCMM 的聚簇质量,即不仅考虑聚簇,同时考虑节点聚簇的优化能力.在图 5 中,平均的聚簇偏移量是指转发树的聚簇偏移量 CO 与应该被聚簇的节点数量之比.图 5 显示了平均聚簇偏移量

CO 保持与聚簇失败率大体一致的增长趋势,并且其值保持在较低的范围之内.因此,可以推断 TCMM 在聚簇数量和质量上都具备可观的效果.

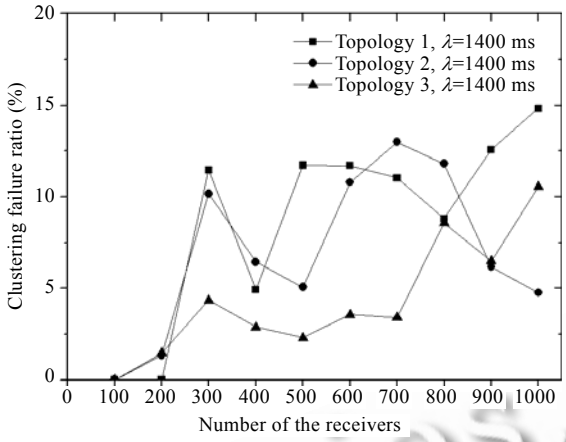

Fig.3 Clustering failure ratio in TCMM

图 3 TCMM 聚簇失败率

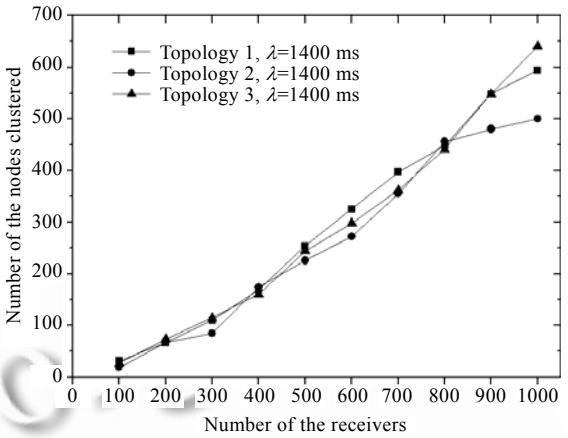

Fig.4 Number of the nodes that are clustered in TCMM

图 4 TCMM 中被聚簇的节点数量

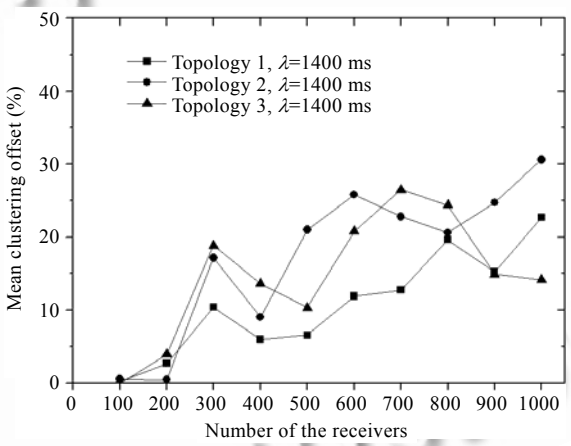

Fig.5 Mean clustering offset in TCMM

图 5 TCMM 中平均聚簇偏移值

从图 3~图 5 还可以看出:在不同拓扑下,相应指标会有所变化,但在各拓扑下,都具备较好的聚簇能力.在下文中,我们将不再重复关于不同拓扑的同样描述.

4.1.2 Internet 实验

在本部分中,$h(i)$表示一个非 leader 节点 i 所对应簇的 leader 节点.特殊情况下,当 i 为 leader 时,$h(i)$=0.定义群组最短聚簇率(scr)为 $scr=\frac{\sum_{i\in R}(d(h,i)/d(s(i),i))^{\frac{1}{2}}}{|C|}$,其中,$C$ 指被聚簇的节点集合,$s(i)$指到 i 最近(度量为延迟)的节点, $R=\{i\,|\,h(i)\neq 0, h(s(i))\neq h(i), i\in C\}$. 从 scr 的定义可以看出,该值越小,其聚簇质量越好.注意,scr 的取值区间为$(0,+\infty)$.

这部分实验用取自 PlanetLab 测试床的 ping 数据集[19]对 TCMM 的聚簇特性和效果进行了验证.我们首先过滤了数据集中的一些无效记录,得到包含 342 个主机的有效 ping 数据集.根据该数据集,我们用 TCMM 聚簇方式对上述 342 个主机进行了 100 次聚簇(每次聚簇采用随机且不相同的加入顺序),并得到相关聚簇特征数据

(见表 1).从表 1 可以看出,在 3 种簇范围阈值λ配置下,TCMM 的聚簇数量和质量都比较高.我们还可以看出,随着λ值的增大,聚簇节点数量会增加,簇数量会下降.因此,λ值增大有益于簇点树收敛.然而,*asr* 值随着λ值的增大而增大.因此,λ值的选择应该使拓扑簇尽可能地覆盖中等规模的网络,但不宜取值过大.

Table 1 Clustering characteristic of TCMM in 100 join sequences

表 1 100 种加入顺序下的 TCMM 聚簇特征

λ (ms)	Min number of clustered nodes	Max number of clustered nodes	Mean number of clustered nodes	Min number of clusters	Max number of clusters	Mean number of clusters	Min *scr*	Max *scr*	Mean *scr*
1	238	252	245.05	98	107	101.53	0.50	0.56	0.53
6	272	284	278.75	80	88	84.56	0.75	0.97	0.86
15	290	301	295.61	68	81	73.10	1.12	1.46	1.28

4.2 其他性能分析

4.2.1 树代价

在该部分实验中,簇范围阈值λ取 1 400ms 和 2 000ms,其中 2 000ms 低于绝大多数不同 transit 域之间的链路延迟.我们用树代价率(tree cost ratio)衡量组播转发树的代价,它是某应用层组播方案构造的树代价与 SPST(shortest path source tree)代价的比值.树代价是指所有与树相关的链路代价之和,其中一个链路的代价为其 Stress 和距离的乘积.

图 6 显示了 TCMM 和 HMTP 在树代价指标方面的比较.由图 6 可以看出,TCMM 在不同拓扑不同参数下所构造的转发树代价明显比对应的 HMTP 树要低.另外,在图 6 中 HMTP 出现了一个明显的异常点(Topology 2 中 400 个接收者处),其树代价比在所有大规模群组下的都要高.出现上述异常点有多种原因,但主要原因是所选的加入顺序对 HMTP 产生了非常不利的负面影响.整体来看,TCMM 树代价随着群组规模的增大平缓地增长,没有明显的异常点.在不同的拓扑下,不同的簇范围阈值λ可能有不同的效果,如图 6 所示.但可以看出,只要λ不选太大的值,TCMM 就能在很大程度上改善转发树的代价.

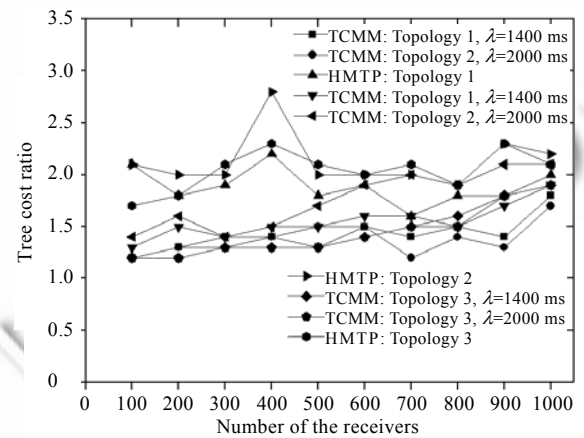

Fig.6 Comparison of tree cost ratio between TCMM and HMTP

图 6 TCMM 和 HMTP 的树代价率比较

4.2.2 Stress

Stress 指标是指同一组播数据包经过某确定链路的次数.在理想的情况下,相关链路的 Stress 为 1.仅有 IP 组播能实现上述理想情况.平均 Stress 值在一定程度上反映了组播质量,即平均 Stress 值越小,链路上的冗余数据包就越少.然而,平均 Stress 值评价无法反映链路的最大(或较大)的 Stress 值情况.

图 7 比较了 TCMM 和 HMTP 在不同群组规模下的平均 Stress 值.其中,Unicast Star 是指用完全单播实现组播功能.平均 Stress 值的大小与拓扑特征有直接关系,因此我们同时画出了一种 IP 组播协议 DVMRP 和 Unicast

Star 的 Stress 情况.在 DVMRP 中,不存在冗余的数据包,故其平均 Stress 值为 1.随着群组规模下的增长,TCMM 和 HMTP 的平均 Stress 值都平缓增大,且在不同群组规模下均保持较低的值.TCMM 将若干节点根据距离相近程度进行聚簇,且一个拓扑簇的成员数量不受限制,可能使某些簇内的平均 Stress 增大,从而可能影响到整个组的平均 Stress 值.因此,在某些情况下,TCMM 的平均 stress 要比 HMTP 的大,如图 7 所示.

图 8 显示了 TCMM 和 HMTP 中较大的 Stress 值的分布情况,其中横轴表示某确定的 Stress 值,纵轴表示具有某 Stress 值的链路数量.由图 8 可以看出,TCMM 和 HMTP 的 Stress 分布曲线都有一个长尾(tail).此外,对于较大的 Stress 值,TCMM 中的相应链路数量明显比 HMTP 中的要少.

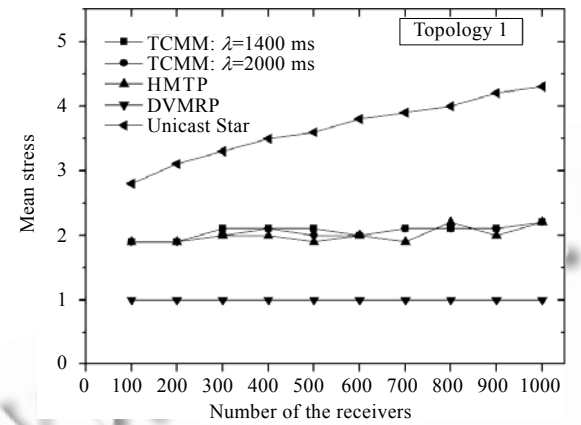

Fig.7 Comparison of mean stress among TCMM, HMTP and DVMRP

图 7 TCMM,HMTP 和 DVMRP 的平均 Stress 值比较

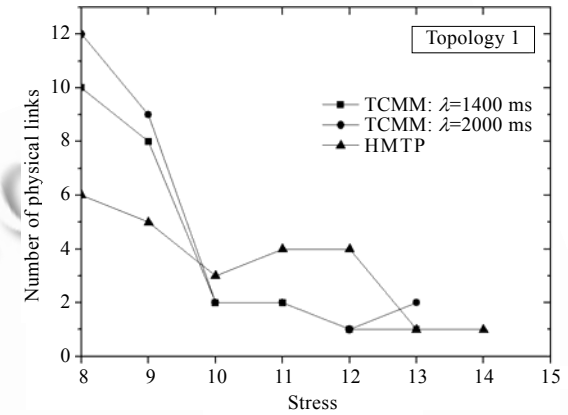

Fig.8 Stress distribution in TCMM and HMTP

图 8 TCMM 和 HMTP 中的 Stress 分布

4.2.3 延 迟

我们用平均延迟率来衡量一个转发树的延迟,如图 9 所示.一个节点的延迟率是指在组播转发树中从数据源到该节点的距离与两点之间最短的单播距离之比.在规模较小的群组中,TCMM 树的平均延迟率比 HMTP 树要大.这是因为在 TCMM 中,尽管某节点的 root path 上可能还有一些节点有剩余的扇出度,但其仍然从相近的节点接收数据.然而,随着群组规模的不断增大,多数节点的扇出度达到饱和,TCMM 的聚簇优势越来越明显,TCMM 树的平均延迟率也由此低于对应的 HMTP 树的平均延迟率.由图 9 还可以看出,TCMM 树的平均延迟率有明显的随着群组规模增大而下降的趋势,而 HMTP 则表现出不规则的变化.

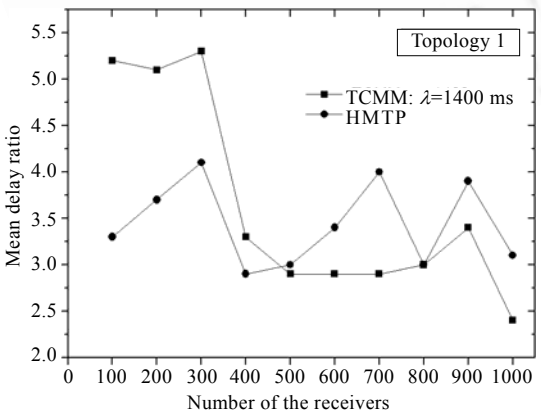

Fig.9 Comparison of mean delay ratio between TCMM and HMTP

图 9 TCMM 和 HMTP 的平均延迟率比较

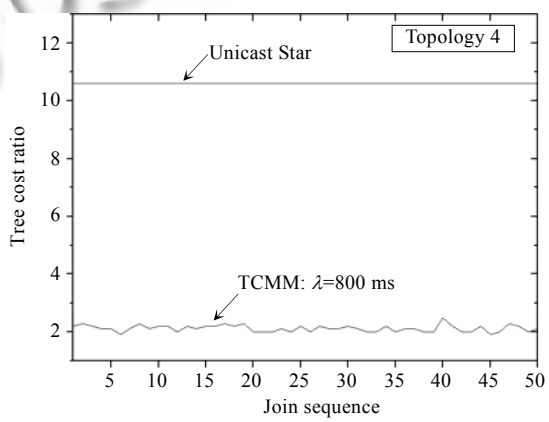

Fig.10 Variation of tree cost ratio in different join sequence in TCMM

图 10 不同加入顺序下 TCMM 的树代价率变化情况

4.2.4 对不同加入顺序的适应能力

我们用一个包含 1 000 个接收者的群组来验证 TCMM 对不同加入顺序的适应能力,如图 10 所示.其中,横轴表示 50 种随机生成的加入顺序.为了反映 TCMM 树代价的优劣,图 10 也给出了 Unicast Star 在相同拓扑下的代价.从图中可以清楚地看到,TCMM 树代价在不同加入顺序下仅作微小变化,表明了 TCMM 具有很好的适应不同加入顺序的能力.具体原因可参见上文定理 1 和定理 5.

5 结 论

本文提出了一种应用于应用层组播的具有拓扑感知能力的聚簇模型,并在此基础上提出了一种应用层组播方案 TCMM.根据节点间的相近特征,TCMM 实现了对成员节点的聚簇,从而实现了本地化的数据包传输.因此,TCMM 能够获得较好的组播性能.此外,拓扑感知的聚簇使得 TCMM 能够在不同的加入顺序下构建较高质量的组播转发树.实验结果分析进一步表明,TCMM 能够有效地实现拓扑感知的聚簇,并具备良好的组播性能.

References:

[1] 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.

[2] Shi SY, Turner JS, Waldvogel M. Dimensioning server access bandwidth and multicast routing in overlay networks. In: Nieh J, ed. Proc. of the 11th Int'l Workshop on Network and Operating Systems Support for Digital Audio and Video. New York: ACM Press, 2001. 83−91.

[3] Malouch NM, Liu Z, Rubenstein D, Sahu S. A graph theoretical approach to bounding delay in proxy-assisted, end-system multicast. In: Liebeherr J, Gross T, eds. Proc. of the International Workshop on Quality of Service (IWQoS 2002). Piscataway: IEEE Computer Society Press, 2002. 106−115.

[4] Cao J, Lu SW. A minimum delay spanning tree algorithm for the application-layer multicast. Journal of Software, 2005,16(10): 1766−1773 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/16/1766. htm [doi:10.1360/161766]

[5] Zhang BC, Jamin S, Zhang LX. Host multicast: A framework for delivering multicast to end users. In: Proc. of the IEEE INFOCOM 2002. Piscataway: IEEE Computer Society Press, 2002. 1366−1375.

[6] Banerjee S, Bhattacharjee B, Kommareddy C. Scalable application layer multicast. Computer Communication Review, 2002,32(4): 205−220. [doi: 10.1145/964725.633045]

[7] Chu YH, Rao SG, Zhang H. A case for end system multicast. IEEE Journal on Selected Areas in Communications, 2002,20(8): 1456−1471. [doi: 10.1109/JSAC.2002.803066]

[8] Mathy L, Canonico R, Hutchison D. An overlay tree building control protocol. In: Crowcroft J, Hofmann M, eds. Proc. of the 3rd Int'l Workshop on Networked Group Communication Networked Group Communication. Berlin, Heidelberg: Springer-Verlag, 2001. 76−87.

[9] Castro M, Druschel P, Kermarrec AM, Rowstron A. SCRIBE: A large-scale and decentralized application-level multicast infrastructure. IEEE Journal on Selected Areas in Communications, 2002,20(8):100−110.

[10] Tran DA, Hua KA, Do TT. ZIGZAG: An efficient peer-to-peer scheme for media streaming. In: Bauer F, ed. Proc. of IEEE INFOCOM 2003. Piscataway: IEEE Computer Society Press, 2003. 1283−1292.

[11] Banerjee S, Kommareddy C, kar K, Bhattacharjee B, Khuller S. Construction of an efficient overlay multicast infrastructure for real-time applications. In: Bauer F, ed. Proc. of the IEEE INFOCOM 2003. Piscataway: IEEE Computer Society Press, 2003. 1521−1531.

[12] Lao L, Cui JH, Gerla M, Chen SG. A scalable overlay multicast architecture for large-scale applications. IEEE Trans. on Parallel and Distributed Systems, 2007,18(4):449−459. [doi: 10.1109/JSAC.2002.803066]

[13] Zhao Q, He Y, Zhang JZ. A hybrid approach for overlay multicast. In: Ni J, ed. Proc. of the 1st Int'l Multi-Symp. on Computer and Computational Sciences. Alamitos: IEEE Computer Society Press, 2006. 496−502.

[14] Li XL, Striegel AD. A case for passive application layer multicast. Computer Networks, 2007,51(11):3157−3171. [doi: 10.1016/j.comnet.2007.01.016]

[15] Tan SW, Waters G, Crawford J. A performance comparison of self-organising application layer multicast overlay construction techniques. Computer Communications, 2006,29(12):2322–2347. [doi: 10.1016/j.comcom.2006.02.020]

[16] Ng TSE, Zhang H. Predicting Internet network distance with coordinates-based approaches. In: Proc. of the IEEE INFOCOM 2002. Piscataway: IEEE Computer Society Press, 2002. 170–179.

[17] Zegura EW, Calvert KL, Bhattacharjee S. How to model an Internetwork. In: Proc. of IEEE INFOCOM'96. Piscataway: IEEE Computer Society Press, 1996. 594–602.

[18] The Network Simulator-ns2. 2008. http://www.isi.edu/ns-nam/ns

[19] PlanetLab. 2004. http://pdos.csail.mit.edu/~strib/pl_app

附中文参考文献:

[4] 曹佳,鲁士文.应用层组播的最小延迟生成树算法.软件学报,2005,16(10):1766−1773. http://www.jos.org.cn/1000-9825/16/1766.htm [doi:10.1360/161766]

张新常(1975－),男,山东泰安人,博士,助理研究员,主要研究领域为下一代互联网络,组播技术.

罗万明(1973－),男,博士,副研究员,CCF高级会员,主要研究领域为下一代互联网络,网络协议.

王正(1979－),男,博士,主要研究领域为下一代互联网络.

阎保平(1950－),女,博士,研究员,博士生导师,主要研究领域为大规模资源定位与寻址技术,下一代互联网技术.