

一种减轻 Xcast 网络路由节点负载的动态规划策略

秦继林¹, 郑明春^{1,2}, 吴春婧¹

(1. 山东师范大学, 山东 济南 250014; 2. 中国科学技术大学, 安徽 合肥 230026)

摘 要: 基于显式组播(Xcast)机制, 介绍了一种减少组播组中存放自动转发信息的状态节点数量, 从而减轻路由路径负担的策略。通过算法描述和分析及实验仿真, 此方案可以有效地节省网络资源。

关键词: 显式组播; 最小化; 自动转发信息

中图分类号: TP393.01

文献标识码: A

文章编号: 1673-629X(2006)08-0038-03

Dynamic Programming for Easing State Nodes in Xcast

QIN Ji-lin¹, ZHENG Ming-chun^{1,2}, WU Chun-jing¹

(1. Shandong Normal Univ., Jinan 250014, China; 2. China Univ. of Sci. and Techn., Hefei 230026, China)

Abstract: In this paper, propose a mechanism which based on explicit multicast. It can minimize the number of routers with forwarding states to free the routers in multicast trees. From the algorithm and its test, the mechanism can save the resource of network effectively.

Key words: explicit multicast; minimization; forwarding state

0 引言

组播技术通过提供单点到多点, 多点到多点的通信方式, 有效提高了网络的利用率, 大大节省了网络带宽, 使得应用有了很好的可扩展性。因此, 组播在视频会议, 协同工作以及实时信息发布等新型网络中发挥了巨大的作用^[1]。组播路由问题也逐渐成为网络资源优化问题的研究热点。

传统的 Internet 组播模型是主机(Host)组模型^[2]。组员由 D 类 IP 地址标记。部分 D 类地址(从 224.0.0.0 到 224.0.0.255)被保留作为永久的组地址。文献[3]中提到了一种源特定组播模型, 是用组播通道代替了单点到多点的连接。在这两种组播方案的路由机制中, 最短路径树中路由由路径上的节点都必须存放组或通道的转发信息。此信息决定其在组播树中的邻接节点, 转发信息的 ID 是组地址或是通道标记。这样就造成节点负担过重的现象。因此, 为了合理而有效地利用网络资源, 要在保证分组传输效率的前提下, 尽可能减少节点信息量, 从而减轻路由节点的负担, 使节点资源得到有效利用。

1 相关工作

目前已经提出很多减轻路由节点负载的方法。与文

中相关的有两种: 其一是用单个组播树传送拥有相似接收者的多个组播树的数据^[4~7]。但是, 用这种方案, 某些接收者可能从一个其非所属的组播组中收到不必要的脏数据; 另一种是显式组播(Xcast, Explicit Multicast), 它是用单播平台来转发组播数据的^[5]。Xcast 支持相当多数量的少量多点传送, 通过显式对数据包中的目标文件进行编码实现, 而非通过组播地址完成。这种方法不要求组播树的任何中介路由由节点存放转发信息, 而把组中所有接收者的地址存放在每个 Xcast 数据包头中, 中间节点只需简单地根据包头中的路径信息来决定下一跳, 并转发数据包, 节点负载相对减轻。但是, 在组播组接收者很多的情况下, 数据包头的信息量就会增大, 包处理的延时也会相应增加, 这种方法就不太适用了^[5]。

事实上, 一个组播组的接收者数量是随时变化的, 因为主机可能随时加入或是离开组播组。当接收者多时和少时所适用的数据转发方案可能不同, 而方案切换又可能会导致服务中断。所以, 文中所研究的方案要兼顾组播组的个数和组中接收者的数量。

2 减轻节点负载策略研究

2.1 网络模型

笔者研究组播的时候, 通常是把网络表示为一个带权有向图 $G(V, E)$, 其中 V 表示节点集合, 节点可以是主机也可以是组播路由器; E 表示连接节点的通信链路集合, 每条链路都有一些相关参数描述其当前状态。而组播组则是图中的一个子集, 这个子集根据网络拓扑结构和当前的网络状态构建一棵组播树, 通常是一棵前向最短路径树。

收稿日期: 2005-11-24

基金项目: 山东省优秀中青年科学家科研奖励基金(304068)

作者简介: 秦继林(1981-), 女, 内蒙古人, 硕士研究生, 研究方向为 P2P, 网络层及应用层组播; 郑明春, 教授, 研究方向为计算机网络应用、网络拥塞控制。

数据由根节点传送到每个接收者。对于单点到多点的通信,组播树的根节点是发送者;对于多点到多点的通信,根节点是转播节点(类似于 PIM-SM 中的 RP^[8])。这样,每个接收者都是叶子节点。

2.2 策略描述

在动态规划最优性定理的思想基础上,研究如何合理地最小化每棵组播树中的状态节点数量的方案。它是先从叶子节点到根节点查询尽可能少的状态节点,再从根到叶子节点指派状态节点。这种方案可以快速地使资源得到优化配置。

在上文提到的显式组播(Xcast)中,当某个接口下游的状态节点和接收者增加的时候,数据包头中的目的节点地址就会相应增加,每个非状态节点就会去查询更多的地址,延时相应加大,进而影响整个网络的工作效率。因此,要对目的节点数加一个限制,即目的节点数最大值(这里用 δ 表示, $\delta \geq 1$)。显然,根节点即状态节点,叶子节点为非状态节点。图 1 就是文中方案的一个例子。

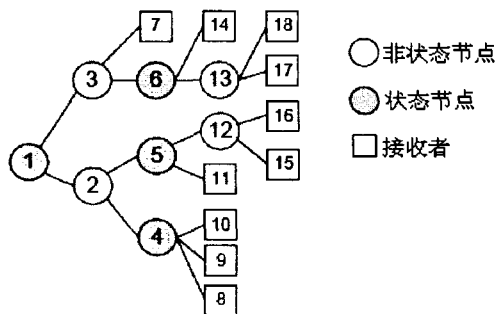


图 1 基于 Xcast 的动态规划策略模型

节点 1, 4, 5 和 6 是组播树中的状态节点; 4 和 5 是节点 1 从下游接口到节点 2 的下游状态节点; 节点 15 和 16 则是节点 5 从其接口到节点 12 的下游接受者; 节点 1 是节点 6 的上游状态节点。节点 1 转发数据给(4, 5), (6, 7); 节点 4 转发数据给(8), (9), (10); 节点 5 转发数据给(11), (15, 16); 节点 6 转发数据给(14), (17, 18)。

可见,文中所介绍的策略不像原始的 Xcast 那样,中间节点不存放任何状态信息,只是一味地加大数据包头的信息量;也不像传统的组播路由机制那样,加重中介节点的负担。它对状态信息作了合理而有效的分配,很好地减轻了节点负载。

2.3 算法描述

引理: 给定树 (V_t, E_t) , 对于 $\forall m \in V_t - \{r_t\}$, \exists 最佳策略: $\tau'_m(j_{m, \text{opt}}) = \tau'_{m, \text{opt}}(j_{m, \text{opt}})$, 其中 $j_{m, \text{opt}}$ 是优化策略中 m 的父节点从其下游接口到 m 的接收者的数量; $\tau'_{m, \text{opt}}(j_{m, \text{opt}})$ 是当 m 的父节点从下游接口到 m , 有 $j_{m, \text{opt}}$ 个接收者时, m 的子树中状态节点的数量; r_t 是根节点。以上用归纳法很容易求证, 这里就不再证明了。

2.3.1 具体算法

给定一棵树 (V_t, E_t) ,

查询 V'_{\min} 和 σ'_m :

● 第一步, 初始化。

当 j 为 1 时, $\tau'_m(j)$ 置 0; 当 $1 \leq j \leq \delta$ 时, $\tau'_m(j)$ 置 ∞ 。并从 1 到 $|V_t|$ 标记每个节点。

● 第二步, 查询状态节点数的最小值。

for m from $|V_t|$ to 2, (m 不是根节点)

if $(|c'_m| = 1), n \in c'_m$, then $\tau'_m(1) = \tau'_n(1)$;

if $(|c'_m| > 1)$, then

$$\tau'_m(1) = 1 + \sum_{n \in c'_m} \min_{1 \leq j \leq \delta} \{\tau'_n(j)\};$$

for j_m from 2 to δ , if 集合

$$\{j_n: n \in c'_m, 1 \leq j_n \leq \delta, \sum_{n \in c'_m} j_n = j_m\} = \emptyset, \text{ then}$$

$$\tau'_m(j_m) \leftarrow \infty;$$

else

$$\text{赋值 } \min_{\{j_n: n \in c'_m, 1 \leq j_n \leq \delta\}} \{\sum_{n \in c'_m} \tau'_n(j_n) | \sum_{n \in c'_m} j_n = j_m\} \text{ 给 } \tau'_m(j_m);$$

end if;

end for;

end for;

$$V'_{\min} \leftarrow 1 + \sum_{n \in c'_m} \min_{1 \leq j_n \leq \delta} \{\tau'_n(j_n)\};$$

● 第三步, 寻找优化的状态节点分配方式。

$$j'_n \leftarrow \arg \min_{1 \leq j_n \leq \delta} \{\tau'_n(j_n)\}, n \in c'_t;$$

for m from 2 to $|V_t|$, m 非根节点;

if $(|c'_m| = 1 \ \& \ j'_m = 1)$ then

$$\sigma'_m = 0, j'_m = 1;$$

else if $(|c'_m| > 1 \ \& \ j'_m = 1)$, then

else

$$\sigma'_m = 1, j'_m = \arg \min_{1 \leq j_n \leq \delta} \{\tau'_n(j_n)\}$$

$$\sigma'_m = 0,$$

$$\{j'_n\} \leftarrow \arg \min_{\{j_n: n \in c'_m, 1 \leq j_n \leq \delta\}} \{\sum_{n \in c'_m} \tau'_n(j_n) | \sum_{n \in c'_m} j_n = j'_m\};$$

end if;

end for;

算法中的符号注释:

T 代表组播树集合; t 代表一棵组播树; p'_m 代表组播树 t 中 m 节点的父节点; c'_m 代表组播树 t 中节点 m 的孩子节点的集合; r_t 代表组播树 t 的根节点; σ'_m 是一个二元变量, 当 m 是 t 中的状态节点时为 1, 否则为 0; τ'_m 是 m 的上游状态节点; D'_m 为 m 节点从其所有下游节点的目的节点的集合; $D'_{m,n}$ 为组播树 t 中, m 节点从其下游节点到 n 节点的目的节点的集合; $\tau'_m(j)$ ($1 \leq j \leq \delta$) 为 m 节点下游子树的状态节点总数, 这里 j 代表 m 的父节点存放的从其下游节点到 m 节点中的目的节点总数; V'_{\min} 为组播树 t 中状态节点数的最小值。

2.3.2 策略的简单分析与评论

算法在第二步中, 先计算 $\tau'_m(j)$, 其值是基于 m 节点子树的状态节点数最小化而进行选择的; 对于 $j = 1$ 且 $|c'_m| > 1$ 时, m 为状态节点, 否则为非状态节点。根节点不

予考虑,是因为已指定它为状态节点。当状态节点数的最小值得到以后,在第三步中, j_n 是计算得出的 m 的父节点从其下游接口到 m 的接收者的数量。

在组播树 t 中,状态节点数的最小值为 $v'_{\min} = 1 + \sum_{n \in c_m} \tau'_n(j_{n, \text{opt}})$; 由引理可得 $v'_{\min} = 1 + \sum_{n \in c_m} \tau'_n(j_n)$; 其中,要保证 $j_{n, \text{opt}} = \arg \min_{1 \leq j_n \leq \delta} \{\tau'_n(j_n)\}$ 。

由集合 $\{j_n: n \in c_m, 1 \leq j_n \leq \delta, \sum_{n \in c_m} j_n = j\}$ 可以得知,

算法的时间复杂度与 $|V_t|$ 和 $4^{(\delta-1)/2}$ 成正比。可见, δ 值在算法中扮演着重要角色。当 δ 比较小的时候,方案更加优越。由算法结论可以看出,文中所介绍的方案达到了预计的策略优化效果,而且是切实可行的。

2.3.3 仿真

这部分用文献[7,8]介绍过的 Waxman 分布式拓扑网络结构,借用一些小的平面图从不同角度测试了文中的方案。网络中有 80 个节点,70 棵组播树,约有 72 个样本为仿真结果;组播组的大小主要指组中接收者的数量。

图 2 表明组播组大小与 δ 的关系,当 δ 较小的时候,可以减少近一半的状态节点;图 3 显示了组播组大小与亲密度的关系,亲密度越大,接收者的聚集程度越大,状态节点在组中所占比例相对就小,当组的大小接近 90 的时候,由于网络中节点大多具有相似组播组的接收者,不同亲密度的节点将会聚集,从而拥有相近的节点数;图 4 是组播树中状态节点所占比例与包含不同参数的组的大小的关系,参数 α, β 越大,状态节点就会有越多的孩子节点,也就能服务于更多的接收者,组中的状态节点个数也就相应减少。从图中的数据得知,文中所介绍的机制对状态信息作了合理的分配,有效地减少了网络中的状态节点数,很好地减轻了节点负载。

3 结束语

阐述了一种如何合理而有效地分配组播树中状态节点,减轻路由节点负担的方案。通过算法验证和仿真实验,说明此法有其一定的优化效果和可行性,能使网络资源达

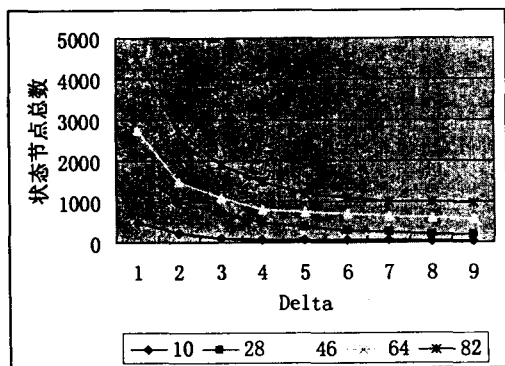


图 2 组播组大小与 δ 的关系

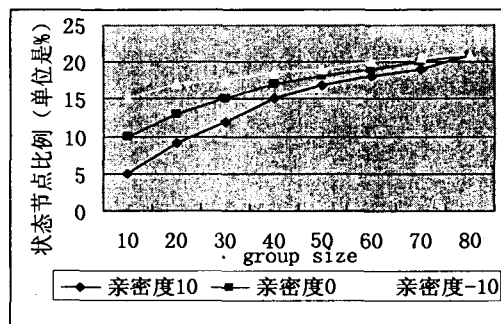


图 3 状态节点数与亲密度的关系($\delta = 2$)

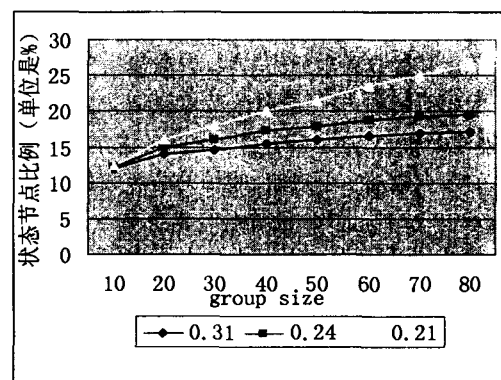


图 4 状态节点所占比例与组播组大小的关系
到更有效的利用。如果在此基础上,能有一种更好的机制,在多个组播组中合理而有效地分配转发信息给路由节点,那么组播通信将会变得更加灵活。

参考文献:

- [1] Cormen T H, Leiserson C E, Rivest P L. Introduction to Algorithms[M]. [s.l.]: MIT Press, 1997.
- [2] Deering S. Host extensions for Multicasting. RFC 1112[S]. 1989.
- [3] Holbrook H, Cain B. Source-specific multicast for IP[J/OL]. IETF Internet Draft. draft-ietf-ssm-arch-04.txt, 2003.
- [4] Radoslavov P I, Estrin E. Exploiting the bandwidth-memory tradeoff in multicast state aggregation[R]. Tech. Rep. 99-697. [s.l.]: USC Computer Science Department, 1999.
- [5] Boivie R, Feldman N, Imai Y, et al. Explicit Multicast basic specification[J/OL]. IETF InternetDraft. draft-ooms-xcast-basic-spec-05.txt, 2003.
- [6] Phillips G, Shenker S, Tangmunarunkit H. Scaling of multicast trees: comments on the Chuang-Sirbu scaling law[A]. ACM SIGCOMM[C]. [s.l.]: [s.n.], 1999. 41-51.
- [7] Waxman B M. Routing of multipoint connections[J/OL]. IEEE Journal on Selected Areas in Communications, 1988, 6(9): 1617-1622. <http://topology.eecs.umich.edu/inet/>.
- [8] Holbrook H W, Cheriton D R. P multicast channels; EXPRESS support for large-scale single-source applications[A]. ACM SIGCOMM[C]. [s.l.]: [s.n.], 1999. 65-77.