

资源利用率最优的应用层组播树构建算法

王太永, 周 刚

(北京航空航天大学软件开发环境国家重点实验室, 北京 100083)

摘 要: 应用层组播(ALM) 树的构建是一个 NPC 问题, 目前已经提出几种启发式组播树构建算法, 各种启发式算法因为考虑因素的制约都存在一定的局限性. 本文根据当前网路链路状态, 通过计算比较选择父节点构建组播树, 提出了在满足最大网络连接带宽、度约束、足够可用性时间的前提下使组播树的资源利用率最优的算法. 通过仿真实验对此算法的数学模型进行验证, 根据组播树的数据吞吐量分析, 与通常组播树构建算法相比, 基于此算法构建的组播树更加合理稳定.

关键词: 应用层组播; 足够可用性时间; 链路延时

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2006) 12A-2376-04

An Optimizing Resource Usage for ALM Tree Building Algorithm

WANG Tai-yong, Zhou Gang

(National Laboratory of Software Development Environment, Beihang University, Beijing 100083, China)

Abstract: The building of application layer multicast tree is a NP complete problem. Today there are several heuristic ALM tree building algorithms. But all the algorithms have some restrictions because of their considering conditions. Making good use of the ALM tree resource, it selects its parent by comparing network link states to build the ALM tree. Considering of the highest network bandwidth, degree constrained, enough living time, we present an algorithm based on Optimizing resource usage for building ALM tree to get a big throughput. This algorithm is shown to be more stable and more balanced for building the ALM tree by simulation of the mathematical model comparing to other ALM tree building algorithms.

Key words: application layer multicast; enough living time; link delay

1 引言

组播是一种高效的数据传输方式, 传统 IP 组播对路由器要求高、组播加入退出开销大、安全拥塞控制机制复杂, 因此不利于在 Internet 上部署. 应用层组播中数据的路由、复制、转发由成员主机完成, 有效的弥补了 IP 组播存在的缺点. 应用层组播保持了网络的“单播、尽力转发”模型, 尤其适合实时流媒体数据的传输. 它将流媒体数据信息传输的特点与其“时间上集中、空间上分布”的特点相结合, 主要应用于视频会议、远程教育、在线电视直播、视频点播等领域. 代表性的研究项目主要有 End System Multicast、ALMI、NICE、Zigzag、CoopNet、Overcast、Yoid 等^[1].

网络中所有主机根据获取的数据源不同构成不同的组播组, 通常一个组播组的成员构成一棵应用层组播树, 组播组的数据在组播树上转发. 应用层组播树的构建关键在于父节点的选择, 考虑的主要因素有吞吐量和组播树的稳定性, 同时这也成为组播树构建算法好坏的衡量标准. 目前通用的应用层组播树构建算法可归结为随机组播树算法、最短路径树算法、

最大带宽组播树算法等几种^[3]. 本文综合考虑多个因素, 抽象出一个更全面的数学模型, 提出了一种基于资源利用率最优的组播树构建算法.

2 应用层组播树算法比较

在实际网络中, 各个节点间的链路带宽和传输延时各不相同, 组播树的构建受到各种因素的制约, 考虑不同的因素可以构建不同的组播树算法. 表 1 对通用的组播树构建算法随机组播树算法、最短路径树算法、最大带宽组播树算法的优缺点及算法时间复杂度进行了比较.

表 1 应用层组播树算法对照表

算法	随机组播树	最短路径树	最大带宽组播树
优点	易于实现, 无需前期链路测量	组播树吞吐量大, 网路资源消耗少	组播树吞吐量大
缺点	不考虑节点性能影响服务质量	考虑因素简单, 性能无法保证	链路带宽不易测量, 组播树不稳定
时间复杂度	$O(1)$	$O(n^2)$	$O(n^2)$

为了提高网络资源利用率, 建立一棵相对稳定的应用层

组播树,资源利用率最优的应用层组播树构建算法选择发送父节点时,在满足最大网络连接带宽、度约束、足够可用性时间的前提下使组播树的资源利用率最优,这样构建的组播树相对稳定,数据吞吐量较大。此算法根据当前网络状态下各节点的实际网络状态选择发送父节点,充分有效的利用了各个节点的服务能力,并且使此组播树在全局范围内收敛到最优而不是局部范围内。

3 资源利用率最优的应用层组播树构建问题的数学模型

在应用层组播树的构建过程中,通常将整个覆盖网络看作一个连通无向图 $G = (V, E)$, 其中 V 表示网络中所有节点的集合,每个节点代表一台主机; E 表示网络中节点构成的所有链路集合。通过计算 E 中链路上传输的数据量,可计算组播树的资源利用率 T_R 。我们考虑当前组播树共有 n 个节点,第 j 个节点出度为 d_j , 节点最大出度数为 $D_{\max}$ 。如果某个节点 i 要加入此组播树,它会向 G 中的候选父节点发送测试数据,通过测试链路带宽 $B(i, j)$ 、足够可用性时间 $T_{i,j}$ 等因素选择一个节点作为发送父节点。新节点加入组播树后的连通无向图为 $G' = (V', E')$ 。

定义 1 ALM 资源利用率: 用于表示组播树有效资源利用情况的指标,表示为通过组播树传输的子节点接收到的数据量与父节点发送的数据量的比值。如图 1 所示,在父节点 B 和子节点 E 构成的链路中,实线表示 E 接收到的数据量,虚线表示 B 发出但 E 未接收到的数据量。

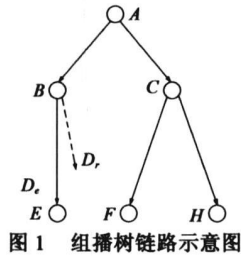


图 1 组播树链路示意图

$$T_R = \frac{\text{子节点接收到数据量}}{\text{父节点发送的数据量}} = \frac{\sum_{e \in E} p_e}{\sum_{e \in E} (D_e + D_r)}$$

其中 $\sum$ 符号表示所有组播树链路求和, D_e 表示链路 e 上子节点接收到的数据量, D_r 表示链路 e 上父节点发送后子节点未接收到的数据量。

定义 2 节点 i 选择发送父节点 j 策略: 需满足以下条件

- (1) 满足度约束 $d_j \leq D_{\max}(j)$
- (2) 具有最大网络连接带宽 $B(i, j) = \max_{0 < k < n} (B_{i,k})$
- (3) 满足足够可用性时间 $T_{i,j}, T_{i,j} = \frac{\text{传输数据大小}}{\text{链路传输带宽}} \leq \text{数据生存时间} = T_{\text{live}}$

数据生存时间 = T_{live}

如果节点不受度约束的限制,同一节点对数据的复制转发次数过多,会导致传输延迟,降低资源利用率;选择传输带宽大的链路,数据传输时间短,会增加数据吞吐量;而足够可用性时间能够保证数据包按时到达目的节点,这样就会减小数据丢弃和重传的概率。综合考虑以上三个因素能够大大提高组播树的资源利用率。

ALM 资源利用率最优的组播树构建算法数学模型

本算法在满足定义 2 选择发送父节点策略的条件下,构造一棵包含 G 中所有节点 V 使 T_R 最大的应用层组播树,其

数学模型公式表示为式(1):

$$\begin{cases} \max_{0 < i, j < n} (T_R) \\ d_j \leq D_{\max}(j), \max(B_{i,j}), T_{i,j} \leq T_{\text{live}} \end{cases} \quad (1)$$

定理 数学模型式(1)可归结为

$d_j \leq D_{\max}(j), \min(t_{i,j}), T_{i,j} \leq T_{\text{live}}$ (其中表示节点 i, j 之间的链路延时) (2)

证明: 由式(1)易推出式(3),求网络链路带宽的最大值可近似归结为求其链路延时的最小值。

$$\Rightarrow \begin{cases} \max_{e \in E'} \frac{\sum_{e \in E'} p_e}{\sum_{e \in E'} (D_e + D_r)} \\ d_j \leq D_{\max}(j), \min(t_{i,j}), T_{i,j} \leq T_{\text{live}} \\ \Rightarrow \max_{e \in E} \frac{\sum_{e \in E} p_e + D_e'}{\sum_{e \in E} (D_e + D_r) + D_e' + D_r'} \\ d_j \leq D_{\max}(j), \min(t_{i,j}), T_{i,j} \leq T_{\text{live}} \end{cases} \quad (3)$$

(D_e', D_r' 分别表示节点 i, j 之间要传输的数据和丢弃的数据)

在当前网络状态下,组播树传输的数据短时间内可近似为常量,要使下一步组播树的资源利用率达到最优,则只与 D_e', D_r' 有关,即只需 i, j 之间丢弃的数据量最小,这样当前步骤下组播树的资源利用率就与 i, j 节点之间的链路延时成反比。因此,数学模型式(1)可归结为式(2)。

4 资源利用率最优的组播树构建算法描述

INPUT: 图 $G = (V, E)$, G 中每个节点 V 的 $D_{\max}$, 根节点 R , 当前预加入的节点为 i , 汇聚点主机 RP (表示为请求加入的节点提供候选父节点的主机)。

OUTPUT: 应用层组播树 T 。

DATA STRUCTURE: 汇聚点主机的节点路由表, 候选父节点集合 PS 。

初始化待求组播树 $T = \phi$, 将根节点 R 加入树 T 中。

(1) 当组播树节点个数 $n = 1$ 时, 此时组播树只有一个根节点 R , 因此加入节点直接选择根节点 R 为其父节点并将此节点加入 T 。

(2) 当组播树节点个数 $n > 1$ 时, 首先节点 i 向汇聚点主机 RP 发出加入组播树请求。

(3) 汇聚点主机 RP 收到加入请求后根据其存储的组播树节点路由表向节点 i 发送候选父节点列表集合 PS 。

(4) 节点 i 向收到的候选父节点发送链路延时检测消息, 同时从 PS 中选出节点度数未达 $D_{\max}$ 的节点。

(5) 节点 i 根据链路延时检测消息对 PS 中的节点按链路延时大小排序, 依次选出延时最小的节点 j , 同时判断其足够可用性时间 $T_{i,j}$ 是否满足 $T_{i,j} \leq T_{\text{live}}$ 。

(6) 如果满足则找到其父节点, 将节点 i 加入集合 T , 然后 i, j 建立连接进行数据传输通信。否则将转到步骤(5)。

(7) 重复以上步骤对欲加入组播树的节点依次进行处理。

此算法的时间复杂度由链路延时时间比较策略决定, 因此最坏情况下选择父节点共要进行 $n \times n$ 次比较, 则其时间复杂

度为 $O(n^2)$, 通常情况下此算法的时间复杂度为 $O(n\log n)$. 这与前面描述的算法时间复杂度相比, 有了很大的提高.

5 算法仿真实验及分析

通过吞吐量和组播树的稳定性对此算法进行衡量, 其中吞吐量用每个节点各个时间的带宽表示; 稳定性可以通过节点带宽随时间变化情况来表示.

试验环境: 在 30 台主机上部署系统对最大带宽组播树构建算法和资源利用率最优的组播树构建算法分别进行测试, 最终得出了以下的分析结果, 其主要的测试环境为: 数据源最大可下载带宽为 4096kbps, 主机节点端的最大上传带宽为 600kbps, 视频播放帧率为 8fps, 共进行 10 次实验, 取一次实验中一台主机数据吞吐量数据, 其变化情况如图 2, 3 所示: (其中一个时间单位为 30 秒).

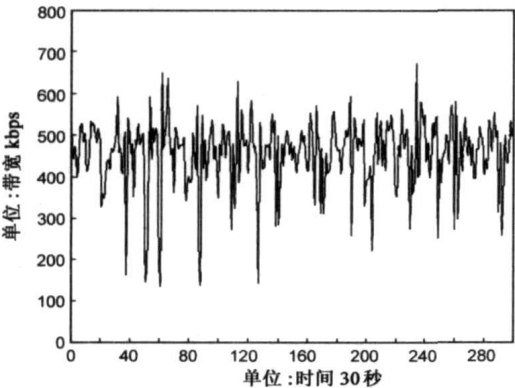


图 2 最大带宽算法吞吐量变化图

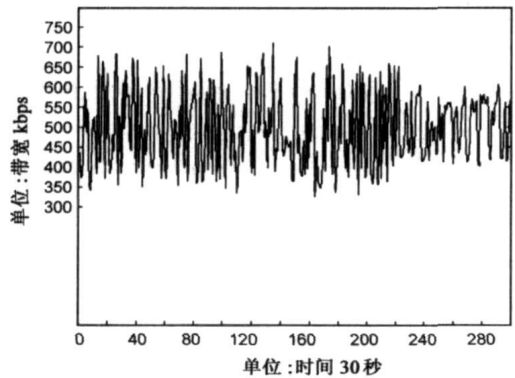


图 3 资源利用率最优算法吞吐量变化图

对 10 次实验得到的数据吞吐量作统计分析得出表 2 所示结果.

通过比较可以看出, 最大带宽组播树构建算法平均吞吐量与资源利用率最优算法相差不是很大, 但前者幅度变化值较大, 原因是经常更换父节点, 组播树不稳定, 整棵组播树的资源利用率低, 而后者幅度变化较小, 标准差小, 组播树相对稳定的多, 原因是每次根据当前网络情况考虑多个因素选择或更换父节点, 因此构建的组播树更加稳定, 资源利用率更高.

从 10 次实验中随机选择两次, 两次实验节点在前 60 分

钟时间内平均更换父节点次数变化情况如图 4 所示, 通过图可以看出, 两次节点平均更换父节点次数变化率基本相同, 说明依据此组播树构建算法构建的组播树稳定性强.

表 2 两种算法试验数据统计分析表(单位: kpbs)

平均吞吐量		极小值	极大值	均值	标准差	幅度值
实验次数	算法					
1	最大带宽组播树	1	848	124.6	62.18	847
	资源利用率最优	5	318	114.8	24.78	313
2	最大带宽组播树	64	408	128.7	45.39	344
	资源利用率最优	55	273	126.6	30.74	218
3	最大带宽组播树	58	547	142.6	51.87	489
	资源利用率最优	59	390	135.4	39.26	331
4	最大带宽组播树	22	508	141.9	54.73	486
	资源利用率最优	83	349	138	39.97	266
5	最大带宽组播树	0	860	141.3	60.44	860
	资源利用率最优	45	393	138.9	40.28	348
6	最大带宽组播树	25	722	126.4	59.43	697
	资源利用率最优	6	760	104.3	42.74	754
7	最大带宽组播树	0	948	127.5	45.56	948
	资源利用率最优	72	404	140	44.18	332
8	最大带宽组播树	26	468	125.8	60.76	442
	资源利用率最优	83	407	128.7	45.03	324
9	最大带宽组播树	5	902	133.9	56.55	897
	资源利用率最优	31	419	128	45.13	388
10	最大带宽组播树	135	674	456.4	85.68	539
	资源利用率最优	322	713	490.5	97.35	391

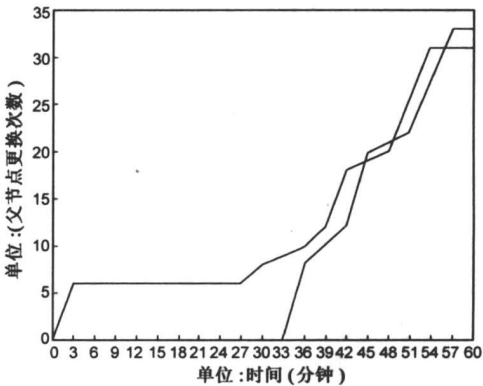


图 4 父节点更换次数变化图

6 结论

资源利用率最优的组播树构建算法在保证应用层组播树吞吐量的基础上, 使构建后的组播树更加稳定, 极大的提高了组播树的资源利用率. 组播树构建算法考虑的因素比较多, 因此各种算法只是在某个方面性能有所改善. 另外由于模拟实验时用到的主机少, 对此算法的性能有待进一步验证.

参考文献:

[1] 章森, 徐明伟, 吴建平. 应用层组播研究综述[J]. 电子学报, 2004, 32(12A): 22- 25.
Zhang Miao, Xu Mingwei, Wu Jiarping. Survey on application layer multicast[J]. Acta Electronica Sinica, 2004, 32(12A): 22

- 25. (in Chinese)
- [2] 胡旺, 李志蜀. 应用层组播时延模型及出度优化算法[J]. 华中科技大学学报, 2005, 33(z1): 76- 88.
Hu Wang, Li Zhishu. Research on the model and algorithms based on out degree optimization for application multicast layer [J]. J Huazhong Univ of Sci & Tech (Nature Science Edition), 2005, 33(z1): 76- 88. (in Chinese)
- [3] 陈阳, 田瑞雄, 李星. 应用层组播树性能的测量研究[J]. 计算机应用研究, 2006, 23(2): 75- 77.
Chen Yang, Tian Ruixiong, Li Xing. Study on performance measurement of application layer multicast trees[J]. Application Research of Computers, 2006, 23(2): 75- 77. (in Chinese)
- [4] Adam Wierzbicki, Robert Szczepaniak, Marcin Buszka. Application layer multicast for efficient peer to peer applications [A]. The Third IEEE Workshop on Internet Applications (WIAPP' 03) [C]. Washington, DC, USA: IEEE Computer Society, 2003. 126- 130.
- [5] Eli Brosh, Yuval Shavitt. Approximation and heuristic algorithms for minimum delay application layer multicast trees [A]. Twenty third Annual Joint Conference of the IEEE Computer and Communications Societies [C]. Hongkong: IEEE INFOCOM 2004, 2004. 2697- 2706.
- [6] Jurcik P, Hanzalek Z. Construction of the bounded application layer multicast tree in the overlay network model by the integer linear programming [A]. Emerging Technologies and Factory Automation, 2005 ETFA 2005 [C]. Catania, Italy, 2005. 503- 510.
- [7] Tsuchiya T, Yoshinaga H, Koyanagi K. STARCcast: streaming collaboration platform using the overlay technology [A]. Applications and the Internet Workshops, 2006 [C]. Phoenix, Arizona, USA: SAINT Workshops 2006. 2006. 23- 27.

作者简介:



王太永 男, 1982 年生于济南, 北京航空航天大学软件开发环境国家重点实验室硕士研究生. 研究方向为 P2P 流媒体、应用层组播.
E-mail: wangty@nlsde.buaa.edu.cn

周 刚 1974 年出生于郑州, 北京航空航天大学博士研究生, 研究方向为 P2P 流媒体, 网络管理.