

文章编号: 1001 - 9081(2006)11 - 2561 - 03

应用层组播树分裂恢复算法研究

王德志¹, 余镇危¹, 甘金颖², 王德毓²

(1. 中国矿业大学(北京)机电与信息工程学院, 北京 100083;

2. 华北科技学院电子信息工程系, 北京 101601)

(wangdz20017@sohu.com)

摘要:应用层组播树是由终端用户构造而成,其稳定性不能得到保证。由于终端用户失效导致组播树分裂,产生多个分支,因此需要快速恢复组播树,保证各组用户接收到组播数据。分析组播树分裂恢复特点,构造了带有时延和度约束的组播树分裂恢复问题数学模型。利用拉格朗日松弛方法分解此复杂问题,提出一种新的应用层组播树分裂恢复算法。仿真实验表明,此算法具有较快的收敛速度,而且算法稳定。

关键词:应用层组播;时延约束;度约束;拉格朗日松弛

中图分类号: TP393.07 **文献标识码:** A

Maintenance algorithm of application level multicast tree

WANG De-zhi¹, YU Zhen-wei¹, GAN Jin-ying², WANG De-yu²

(1. School of Mechanical Electronic & Information Engineering, China University of Mining and Technology, Beijing 100083, China;

2. Department of Electronics and Information Engineering, North China Institute of Science and Technology, Beijing 101601, China)

Abstract: The application level multicast tree is composed of the end hosts which can not ensure the stability of the multicast tree. The failure of the end host in the multicast tree may cause the split of the multicast tree. Therefore, the multicast tree needs to be rebuilt quickly to ensure that every group of clients in the multicast tree can receive the data. Analyzing the characteristics of the division and renewal of the multicast tree, the maintenance model of the application level multicast tree with the delay and degree constraints was constructed. The complicated multicast tree problem was decomposed by Lagrangian relaxation. A new maintenance algorithm of the application level multicast was advanced. The simulation results show that the algorithm is of higher converging speed, stability and easy operation.

Key words: application level multicast; delay constraint; degree constraint; Lagrangian relaxation

IP组播的研究早在20世纪80年代就已经开始,尽管IP组播概念简单和性能高效,但其自身所固有的缺点,如缺少有效的地址分配机制,需要对现有路由器进行改造等,限制了IP组播的实际部署。因此,研究人员最近几年提出了应用层组播的概念,并成为了研究热点^[1]。应用层组播利用现有的网络作为底层的基础设施为端用户提供组播服务,由端用户在物理网络上自组织成一个虚拟的覆盖网络,组播分组沿着在覆盖网络上构造的组播树进行分发^[2]。与IP组播相比,应用层组播易于部署,解决了地址分配问题,容易实现可靠性,但它的效率不如IP组播,可靠性易受端用户系统稳定性的影响^[3]。如何在由于个别用户节点失效导致组播树分裂时,快速恢复组播树,保证脱离节点收到组播数据是必要的。很多应用层组播研究,如Scattercast Overcast Narda等都是通过构造原始组播树的方法来恢复组播树^[4]。本文提出一种新的组播树恢复算法,通过布置组播服务节点来快速恢复由于个别节点失效导致的组播树分裂问题。

组播数据的复制与转发,如图1(a)所示(实线代表组播树)。由于端用户的不稳定性,当个别节点突然失效时,必然导致组播树的分裂,尤其是当具有较多子节点的关键节点失效时必然对组播树影响很大。如图1中m节点失效,必然产生3个组播树分支分裂(图1中A、B、C为组播树分支)。传统的组播树恢复方法是在现有组播树基础上采用原有构造组播树方法,把组播树分支连接到组播树上。但是由于原有组播树节点的度约束,组播树恢复后一些分裂分支节点的时延约束可能得不到满足,组播树效率降低。图1(a)中(虚线代表恢复分支)a1节点受度约束限制只能连接到b2上,从而导致a2节点的时延约束得不到满足。

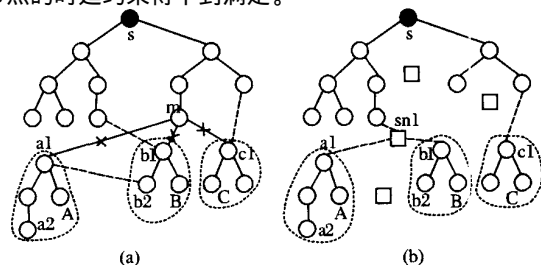


图1 应用层组播树拓扑结构

1 应用层组播树分裂恢复问题及数学描述

1.1 应用层组播树分裂及恢复

应用层组播树是由端用户构成,每一个组播节点都负责

收稿日期: 2006 - 05 - 25 基金项目: 教育部博士点基金资助项目(20030290003)

作者简介: 王德志(1977 -),男,辽宁辽阳人,博士研究生,主要研究方向:计算机网络体系结构、应用层组播; 余镇危(1942 -),男,上海人,教授,博士生导师,主要研究方向:计算机网络体系结构、下一代计算机网络; 甘金颖(1974 -),女,辽宁锦州人,硕士,主要研究方向:计算机控制系统、智能控制; 王德毓(1974 -),男,辽宁辽阳人,工程师,主要研究方向:计算机控制系统、电气自动化。

为了解决上述问题,可以通过布置组播服务节点,让不能满足时延约束的分裂分支节点通过新的组播服务节点重新加入到组播树中,从而保证这些节点的时延约束条件。如图 1(b)所示,节点 a1和 b1连接到新组播服务节点 sn1上,在满足节点度约束的同时也满足时延约束。这些新组播服务节点具有较高的度约束及处理能力,它们的分布可以覆盖所有的分裂分支节点,从而保证组播树恢复后的效率。

1.2 组播树恢复问题数学描述

应用层组播是建立在虚拟覆盖网之上的,而一个覆盖网络可用无向图 $G = (V, E)$ 来表示, V 由组播会话中的源节点和所有目的节点组成, $\forall e \in E$ 为节点间的链路。应用层组播树可以表示为 $T = (V_T, E_T)$, 其中 $V_T \subset V$ 是组播树节点的集合, $E_T \subset E$ 是组播树上链路的集合^[5]。设 $Q \subset V_T$ 是组播树分裂后分裂分支的节点集合,如图 1中 A, B, C区中节点。对于节点 $d \in Q$, 组播源 s 到 d 的路径 $P(s, d)$ 可表示为 $P(s, d) = \{e(i, j) | e(i, j) \in E\}$, 其中 $e(i, j)$ 表示从源 s 到 d 路径上的链路,如图 1(b)中虚线所示的路径与链路。设变量 x_{ij}^d 当 $e(i, j)$ 为源 s 到 d 路径上的链路时为 1, 否则为 0, 从而对于组播树节点存在如下关系:

$$x_{ij}^d = \begin{cases} 1, & \text{当 } e(i, j) \in P(s, d) \\ 0, & \text{当 } e(i, j) \notin P(s, d) \end{cases} \quad (1)$$

$$\sum_{j \in V} x_{ij}^d - \sum_{j \in V} x_{ji}^d = \begin{cases} 1, & \text{当 } i = s \text{ 且 } d \in Q \\ 0, & \text{当 } i \in \{v | v \in V \text{ 且 } v \neq s, v \neq d\} \text{ 且 } d \in Q \\ -1, & \text{当 } i = d \text{ 且 } d \in Q \end{cases} \quad (2)$$

它们表示路径 $P(s, d)$ 上各节点的特性,当路径上节点为源节点时,它只有一条输出链路,为目的节点时只有一条输入链路,而为路径上的中间节点时输入和输出链路数相等。

由于新组播服务节点的加入,增加了组播树代价,这包括新组播服务节点的代价和新添加的链路的代价。因此在考虑如何选取新组播服务节点时,以增加最小组播树代价为目标。同时,因为应用层组播树节点处理能力有限,因此在选取节点时还要考虑节点的度约束问题。为了满足用户应用的需要,端到端的时延约束也是考虑的条件。由于采用新组播服务节点恢复组播树,添加了新的组播链路,因此要对新路径中链路共享的数目进行约束,避免网络负载的集中,平衡网络负载的分配。因此,根据以上论述可以得到应用层组播树分裂恢复问题的数学模型,目标函数为:

$$\text{Min} \left(\sum_{e(i, j) \in E_T} c_{ij} y_{ij} + \sum_{i \in B} h_i z_i \right) \quad (3)$$

其中 c_{ij} 和 h_i 为链路和节点代价; y_{ij} 表示 $e(i, j)$ 为新路径的链路时为 1, 否则为 0; z_i 表示新组播服务节点被选取时为 1, 否则为 0; $B \subset V$ 为备选新组播服务节点。

目标函数的约束条件为:

$$\sum_{j \in V} x_{ij}^d - \sum_{j \in V} x_{ji}^d = \begin{cases} 1, & \text{当 } i = s \text{ 且 } d \in Q \\ 0, & \text{当 } i \in \{v | v \in V \text{ 且 } v \neq s, v \neq d\} \text{ 且 } d \in Q \\ -1, & \text{当 } i = d \text{ 且 } d \in Q \end{cases} \quad (4)$$

$$\sum_{j \in V} y_{ij} \leq u_i z_i, \quad i \in V \quad (5)$$

$$\sum_{e(i, j) \in P(s, d)} de_{ij} x_{ij}^d \leq D, \quad i, j \in V \text{ 且 } d \in Q \quad (6)$$

$$\sum_{d \in Q} x_{ij}^d \leq K y_{ij}, \quad e(i, j) \in P(s, d) \quad (7)$$

其中 u_i 为节点 i 的度约束, de_{ij} 为链路 $e(i, j)$ 的时延,而 D 为路径 $P(s, d)$ 的时延约束, K 为链路共享约束。公式 (5) ~ (7) 分别代表节点度约束、端到端时延约束和链路共享约束。

2 组播树分裂恢复算法

根据上文数学描述可以看出,带约束条件的组播树分裂恢复问题是复杂整数规划问题,是一个 NP 难问题,因此很难直接求出其解。通过分析约束条件可知,约束条件 (4) 是最短路径问题的约束条件,而约束 (5) ~ (7) 是松弛约束条件,因此可以采用拉格朗日松弛方法来解决这种松弛问题。其基本思想就是把复杂问题的一些约束条件引入到目标函数中,从而减少约束条件,降低目标函数的复杂度^[6]。

2.1 组播树分裂恢复问题分解

原问题虽然是整数规划问题,但它具有最短路径问题的约束条件,因此可以利用这种特点,利用拉格朗日松弛方法把它的松弛约束条件 (5) ~ (7) 引入到目标函数中,使原问题转化为最短路径问题,从而易于求解。因此得到新的目标函数为:

$$\text{Min} Z(\lambda) = \sum_{e(i, j) \in E_T} c_{ij} y_{ij} + \sum_{i \in B} h_i z_i + \lambda_1 \left(\sum_{j \in E} y_{ij} - u_i z_i \right) + \lambda_2 \left(\sum_{d \in Q} \sum_{e(i, j) \in E_T} de_{ij} x_{ij}^d - D \right) + \lambda_3 \left(\sum_{d \in Q} \sum_{e(i, j) \in E_T} x_{ij}^d - K y_{ij} \right) \quad (8)$$

约束条件为:

$$\sum_{j \in V} x_{ij}^d - \sum_{j \in V} x_{ji}^d = \begin{cases} 1, & \text{当 } i = s \text{ 且 } d \in Q \\ 0, & \text{当 } i \in \{v | v \in V \text{ 且 } v \neq s, v \neq d\} \text{ 且 } d \in Q \\ -1, & \text{当 } i = d \text{ 且 } d \in Q \end{cases} \quad (9)$$

对目标函数进行整理得:

$$\text{Min} Z(\lambda) = \sum_{d \in Q} \left(\sum_{e(i, j) \in E} (2d_{ij} + 3) x_{ij}^d - 2D \right) + \sum_{e(i, j) \in E_T} (c_{ij} + \lambda_1 - 3K) y_{ij} + \sum_{i \in B} (h_i - \lambda_1 u_i) z_i \quad (10)$$

其中 $\lambda_1, \lambda_2, \lambda_3$ 分别为约束 (5) ~ (7) 对应的拉格朗日乘数。因此目标函数可以被分解为三个子问题函数。

$$P1: \text{Min} Z(\lambda) = \sum_{d \in Q} \left(\sum_{e(i, j) \in E} (2d_{ij} + 3) x_{ij}^d - 2D \right)$$

$$\text{s.t.} \quad \sum_{j \in V} x_{ij}^d - \sum_{j \in V} x_{ji}^d = \begin{cases} 1, & \text{当 } i = s \text{ 且 } d \in Q \\ 0, & \text{当 } i \in \{v | v \in V \text{ 且 } v \neq s, v \neq d\} \text{ 且 } d \in Q \\ -1, & \text{当 } i = d \text{ 且 } d \in Q \end{cases}$$

$$P2: \text{Min} \sum_{e(i, j) \in E_T} (c_{ij} + \lambda_1 - 3K) y_{ij}$$

$$P3: \text{Min} \sum_{i \in B} (h_i - \lambda_1 u_i) z_i$$

其中 $P1$ 为最短路径问题, $P2, P3$ 为无约束条件的求最小值问题,因此原问题被分解为三个不相关的子问题,可以分别对其进行求解,降低算法的难度。

2.2 算法描述

对于子问题 $P1$, 由于是最短路径问题,因此可以采用经典的 Dijkstra 算法,求解出节点 d 经过新组播服务节点到源节点 s 的最短路径。由于 Dijkstra 算法的时间复杂度为 $O(N^2)$, 对于有 $|Q|$ 个节点要加入组播树,每一次迭代都要计算一次最短路径,则算法的复杂度为 $O(|Q|/N^2)$ 。

对于问题 $P2$, 它是一个无约束的求最小值问题。分析公式可知,当 y_{ij} 的系数为负数时,如果 $y_{ij} = 1$ 则这时的解使目标函数最小;而当 y_{ij} 的系数为正数时,如果 $y_{ij} = 0$ 则这时的解使

目标函数最小为 0。因此设 $z_i = c_{ij} + \lambda_j - \mu_j K$, 则有 $P2(\lambda, \mu) = \begin{cases} z_i > 0, y_{ij} = 1 \\ 0, z_i = 0, y_{ij} = 0 \end{cases}$

对于问题 P3, 也是一个无约束的求最小值问题, 根据问题 P2 可得, 设 $z_i = h_i - \lambda_j u_i$, 则有 $P3(\lambda) = \begin{cases} z_i > 0, z_i = 1 \\ 0, z_i = 0, z_i = 0 \end{cases}$

根据拉格朗日松弛理论可知, 对利用拉格朗日松弛方法分解问题求得的解不能够保证是原问题的可行解, 因此要对分解问题求的解进行可行性验证。对于本文算法就是在求得分解后问题 P1P2P3 的解后, 把解代入原问题目标函数及约束条件中, 如果满足就是可行解, 如果不满足条件就要重新计算求解。

对于 λ 的选取, 初始条件下根据经验设定初始值, 然后每次迭代对 λ 值进行修改, 当满足迭代次数或者目标函数值 $Z(\lambda)$ 在规定的步数内变化不超过一个给定的值 ϵ , 这时就认为目标函数值不可能再变化, 可以停止运算。

3 仿真实验及结果分析

本文采用 Matlab 7.0 作为仿真工具, 分别对具有 40 节点和 80 个节点的应用层组播网络进行仿真实验。在仿真环境中, 节点间的链路采用随机生成的方法, 并保证仿真网络为全连通网络。网络链路的时延代价为一个范围的随机整数, 并且每个节点的度约束指定为 $[1, n/2]$ 之间的一个随机整数。

表 1 不同约束条件下目标函数值

节点数	时延度	目标函数	时延度	目标函数	时延度	目标函数
40	10, 3	126	10, 8	116	20, 5	109
	10, 5	121	20, 3	112	20, 8	106
80	15, 5	206	15, 15	185	25, 10	171
	15, 10	194	25, 5	174	25, 15	168

为了评价算法性能, 分别对不同规模的网络在不同时延和度约束条件下求的目标函数值即最小代价进行比较, 如表

(上接第 2560 页)

树结构下, 随着群用户的增长, 中心服务器负载的增长速度远慢于二层 K 叉树的增长速度。

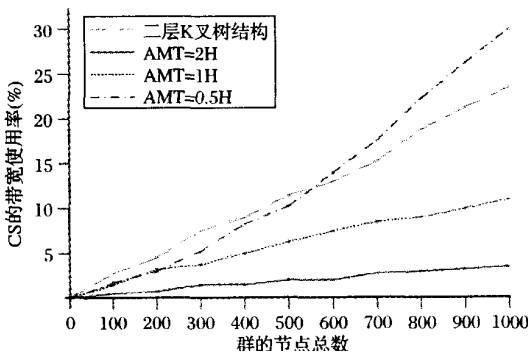


图 8 CS 带宽使用率

5 结语

三层 K 叉树架构有效地将群的转发处理和状态维护分担至群成员节点, 使用双层服务器共同来实现群服务。中心服务器主要维护群的永久信息, 转发功能和状态管理由从群成员节点中选取的群服务器来担当, 大大减轻了中心服务器的

负担, 从而使群具有很强的扩展性。性能分析和仿真表明, 三层 K 叉树下的群具有很强的可靠性与扩展性。

为了发现分裂分支节点规模与所需新组播服务节点数目的关系, 对不同分裂分支节点规模网络进行了仿真, 仿真结果如图 2 所示。从图中可以看出, 随着分裂节点的增多, 所需新组播服务节点数目也在增多, 而应用层组播网络规模越大需要的新组播服务节点数目也就越多。

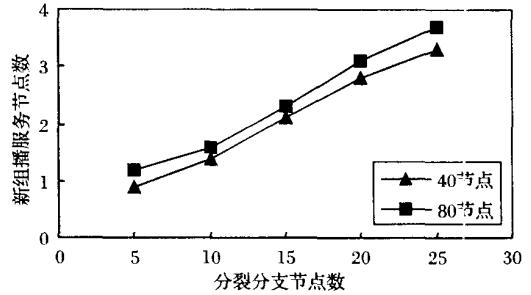


图 2 分裂分支节点与组播服务节点关系

参考文献:

- [1] LU J, SHAN X, REN Y. Overlay networking: applications and research challenges[J]. Journal of Institute of Communication, 2004, 25(12): 46 - 51.
- [2] ZHU Y, WU M, SHU W. Comparison study and evaluation of overlay multicast networks[A]. Multimedia and Expo 2003[C]. 2003, 3: 493 - 496.
- [3] YEO CK, LEE BS, ER MH. A survey of application level multicast techniques[J]. Computer Communications, 2004, 27: 1547 - 1568.
- [4] MATHY L, CANON KO R, HUTEH ISDN D. An overlay tree building control protocol[A]. Proceedings of the Third International Workshop on Networked Group Communication (NGC)[C], 2001, 7: 76 - 87.
- [5] 潘耘, 余镇危, 王励成. 求解应用层组播路由问题的遗传算法[J]. 小型微型计算机系统, 2005, 26(1): 55 - 58.
- [6] 王珩, 王华, 孙亚民. 一种基于拉格朗日松弛的时延约束多播路由算法[J]. 通信学报, 2004, 25(5): 83 - 93.

负载, 从而使群具有很强的扩展性。性能分析和仿真表明, 三层 K 叉树下的群具有很强的可靠性与扩展性。

参考文献:

- [1] CHATTERJEE S, ABHICHANDANIT, LIH, et al. Instant Messaging and Presence Technologies for College Campuses[J]. IEEE Network, 2005, 19(3): 4 - 13.
- [2] MINTZ M, SAYERS A. MSN Messenger Protocol[EB/OL]. http://www.hypothetic.org/docs/msn/MSN_Messenger_Protocol-Home_Page.html, 2006 - 02.
- [3] Jeff_yecn. QQ 协议分析之 TCP/IP 包数据分析[EB/OL]. http://www.cn-paf.net/Class/OtherAnalysis/056912583393970113.htm, 2006 - 03.
- [4] ABAWAJY JH, DANDAMUDI SP. Parallel Job Scheduling on Multicluster Computing Systems[A]. IEEE CNF CLUSTER 2003[C]. 2003, 11 - 18.
- [5] SA NT-ANDRE P. IETF RFC3921, Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence[S], 2004.
- [6] CHU H-T, CHEN W-S, HUANG Y-H, et al. A Novel Design of Instant Messaging Service Extended From Short Message Service With XMPP[A]. Fifth IEEE International Conference on 3G Mobile Communication Technologies[C], 2004, 504 - 508.