第 37 卷 第 19 期
Vol.37 No.19

计 算 机 工 程
Computer Engineering

2011 年 10 月
October 2011

• 网络与通信 •

文章编号：1000—3428(2011)19—0085—03 文献标识码：A 中图分类号：TP393

一种高稳定性应用层组播树构建算法

李午阳，高德远，何 颖，高 翔

(西北工业大学计算机学院，西安 710129)

摘 要：提出一种基于节点在线时间期望的应用层组播树构建算法(MPOT)。根据路径的在线时间期望获得节点的插入位置，节点中断后利用组播节点在线时间的重尾现象，在恢复被迫中断节点时保留节点在线时间信息，构建高稳定性组播树。同时针对断裂恢复问题，提出带一阶预测的 MPOT 算法。模拟结果表明，2 种算法构建的组播树均具有较高的稳定性。

关键词：应用层组播；组播树；重尾；在线时间期望；一阶预测

Construction Algorithm of Application Layer Multicast Tree with High Stability

LI Wu-yang, GAO De-yuan, HE Ying, GAO Xiang

(College of Computer Science and Technology, Northwestern Polytechnical University, Xi'an 710129, China)

【Abstract】This paper proposes a construction algorithm of Application Layer Multicast(ALM) tree based on live-time expectation. The algorithm gets insert node from calculating path-live-time expectation. According to the heavy tails conclusion of online nodes, reserve the live-time information of interrupted nodes in recovering these interrupted nodes do great help to the stability of ALM tree. Aiming at fault recovery, it proposes improved MPOT algorithm with one order prediction. Simulation result shows that the proposed algorithm has better performance compared to the other algorithms.

【Key words】Application Layer Multicast(ALM); multicast tree; heavy tails; live time expectation; one order prediction

DOI: 10.3969/j.issn.1000-3428.2011.19.027

1 概述

组播是 Internet 中点对多点的数据传输方式。IP 组播是在原有 Internet 单播的框架上进行扩展，组播功能主要通过路由器来实现。但 IP 组播在传输技术和管理上存在的问题制约了 IP 组播的发展[1]。

应用层组播技术因其不需要路由器的支持、无需改变原有网络的体系结构等特征所表现出的灵活性而被广泛应用。然而节点可自由退出组播树，在节点失效或退出后其子节点的组播连接被迫中断。因此，如何减少节点退出后带来的传输中断，是应用层组播树稳定性的主要问题。

目前针对组播树稳定性问题的研究主要集中于如何构建低中断频率的组播树，主要算法有基于节点带宽的算法以及基于节点在线时间的算法。文献[2]基于节点带宽的算法基本思想是构造深度较小的组播树来减少节点失效或退出行为的影响；文献[3]基于节点在线时间的算法主要基于组播节点在线时间长度分布呈重尾现象(heavy tails)，即在线时间较短的节点比在线时间较长的节点更容易发生中断或退出行为[4-5]，通过计算组播树中某路径的最小中断期望等方法来获取加入位置，从而减少组播树中断次数；文献[6]提出一种基于节点性能估算的应用层组播算法，根据节点转发能力进行选择插入节点位置。

本文利用节点在线时间的重尾现象，提出一种基于节点在线时间期望的高稳定性应用层组播树构建算法(最大在线时间期望算法)。通过计算各可容纳节点根路径的在线时间期望来获得待插入节点的插入位置。

2 最大在线时间期望算法

对于组播树 $T(V,E)$，从节点 $v_i(v_i \in V(T))$到达组播树 $T(V,E)$ 根节点 $Root$ 的路径称为节点 $v_i(v_i \in V(T))$的根路径 $P(v_i)$。$v_i(v_i \in V(T))$所在根路径 $P(v_i)$的在线时间期望 $E(P(v_i))$，称为节点 $v_i(v_i \in V(T))$根路径的在线时间期望。

2.1 MPOT 算法

为简化说明，定义如表 1 所示的符号。

表 1 符号和术语定义

符号	术语定义
$int(N)$	因节点 N 的中断而受到影响的节点数目
$prob(N)$	在线时间为 t 的节点 N 的中断概率
$desc(N)$	依赖节点 N 转发数据的节点数目

组播树的一个实例如图 1 所示，由于节点 A 负责节点 C 组播数据的转发，因此节点 A 的中断或者退出将导致节点 C 传输中断。那么 $int(A)=1$；由于节点 B、C 没有子节点，因此 $int(B)=int(C)=0$。

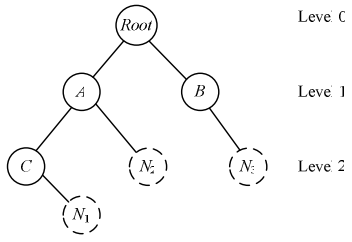

图 1 组播树实例

基金项目：教育部博士点新教师基金资助项目(20070699011)

作者简介：李午阳(1987－)，男，硕士研究生，主研方向：网络与信息安全；高德远，教授、博士生导师；何 颖，硕士；高 翔，副教授

收稿日期：2011-03-21 **E-mail：**alonewolft@gmail.com

根据上述分析，整棵树 $T(V,E)$的期望中断次数可表示为：

$$E(\mathrm{int}(T)) = \sum_{v\in T} prob(N)\times desc(N)$$

由期望中断的定义可以看出，期望中断次数不仅与节点在线时间分布有关，还与节点的子孙节点数量有关。

对于组播树 $T(V,E)$中节点 v_n 的根路径 $v_nv_{n-1}\cdots v_0=P_n$，将一个新节点 v_{new} 插入路径 $v_nv_{n-1}\cdots v_0$ 上的某一个可容纳节点 v_m 后得到的新组播树记为 T_{new}，使 $E(\mathrm{int}(T_{\mathrm{new}}))$ 取得最小值的插入位置是根路径 P_n 上深度最小的可容纳节点。

如图 1 所示，节点 A、B、C 都是可容纳节点，根路径 $P(C)$上有可容纳节点 A、C，其中 A 的深度小于 C 的深度。由上分析可见，节点 C 不是产生最小期望中断的候选插入节点。候选插入节点集合只包含节点 A、B。

根路径的期望在线时间越长，则路径上节点的中断概率就越小。将新节点加入到期望在线时间长的路径上，新节点被中断的概率就会越小。所以，应该在所有根路径的最小深度可容纳节点集合中选取根路径期望在线时间最大的可容纳节点作为插入新节点的位置。

算法的执行步骤如下：

(1)对组播树 $T(V,E)$按深度优先遍历，获得深度最小的可容纳节点集合 $Set(E)$。

(2)对于节点 $N\in Set(E)$，根据其根路径上各节点的概率分布函数 $F_i(x)$计算根路径的概率分布函数 $F(t)$。

(3)对根路径的概率分布函数 $F(t)$计算差分，计算在线时间期望。在线时间期望最大的可容纳节点即为插入位置。

由上分析，对组播树进行深度优先遍历，获得可容纳节点集，可减少可容纳减少结果集合大小，并且在计算在线时间期望时，减少计算次数，很大程度提高了算法性能。与同类算法相比，文献[5]的算法并没有精确考虑节点各个祖先节点的不稳定性问题，而本文算法则通过计算节点所在整条路径的在线时间期望来获得最稳定的可容纳节点，因此，该算法相比同类算法获得的组播树具有更高的稳定性。

2.2 组播树断裂恢复

组播树中节点的中断或退出将导致其子节点或子树断裂，在子树断裂后，子树中各节点会继续申请加入组播树、请求组播服务，这类问题称为断裂恢复问题。由于断裂的子树中的节点在断裂时都已经存在一定的在线时间，由上分析，保留断裂子树中各节点的在线时间信息进行断裂恢复，对于相同节点数量的组播树而言，将提高组播树的稳定性。同时，在实际应用中，应用层组播树中用户节点可自由退出组播树，导致组播树中断现象频繁出现，因此，断裂子树的恢复在实际应用中具有很重要的实际意义。本文提出的 MPOT 算法同样适用于断裂子树的恢复，并且本文针对断裂恢复问题提出了一种改进的 MPOT 算法。

断裂子树的恢复可以分 2 种方式进行：

(1)将子树拆分为单个节点，对节点进行恢复；

(2)直接将断裂子树按原有拓扑结构进行恢复。

对于拆分子树的恢复方法，由于节点中保留了节点在线时间信息，因此直接按照新节点的计算方法进行计算。对于按断裂子树原有拓扑结构恢复的方法，本文称为带一阶预测的 MPOT 算法，考虑了子树根节点加入以后的根路径的在线时间期望，提高了子树的子孙节点的根路径的稳定性。

3 程序模拟及分析

本文利用上述最大在线时间期望算法实现了低中断频率组播树的构建，并进行程序模拟，同时通过与以往同类研究方法进行比较，验证了其正确性及优越性。

实验采用带一阶预测的 MPOT 算法对断裂子树进行恢复，并与以下组播树生成算法进行比较：

(1)最小深度算法(MinDepth 算法)，即选择深度最小的可容纳节点作为插入位置。

(2)随机算法(Random 算法)，即随机选择组播树中可容纳节点作为插入位置。

(3)UBA 算法，即按文献[5]算法选择插入位置。

为便于同以往同类研究进行结果比较，本文采用的参数值与文献[5]中所选参数值保持一致。具体的程序模拟参数值如表 2 所示。

表 2 程序模拟参数

模拟时间/s	根节点容量	新增节点	节点在线时间	节点容量
100 000	10	泊松分布 (λ=1)	对数正态分布 (μ=4.4, σ=1.6)	指数分布 (μ=2)

在线节点数随时间的分布情况如图 2 所示。断裂子树(未拆分)恢复下的累计中断次数曲线如图 3 所示，结果均为 80 次模拟运行的平均结果。

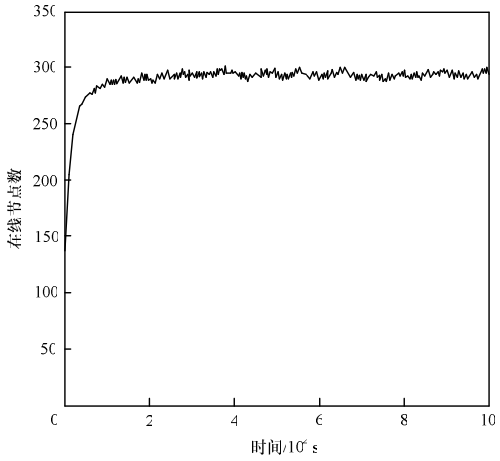

图 2 模拟过程中同时在线的节点数目随时间变化曲线

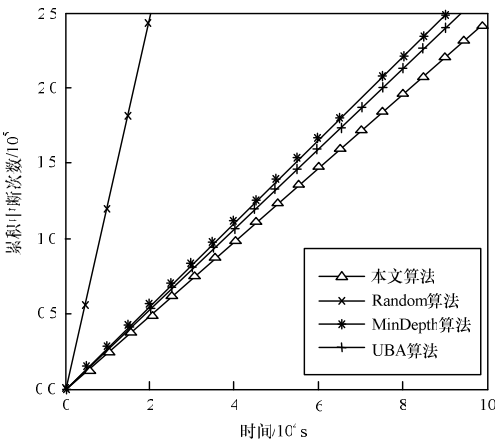

图 3 未拆分断裂子树恢复的累计中断次数曲线

由累计中断次数曲线来看，本文算法结果相比其他算法，中断次数是最小的，并且可以看出随时间增长，中断次数曲线呈线性增长，而本文算法的曲线斜率小于其他算法，因此，可以预计在更长的时间内，其累积中断次数也较小。同时，实验结果表明，拆分断裂子树情况下的累积中断次数曲线也具有类似结果，限于篇幅，不再赘述。由实验结果可以看出，使用本文算法得到的组播树具有更高的稳定性。

采用 UBA 算法与 MPOT 算法累计中断次数变化的对比

曲线如图 4 所示。可以看出，在恢复断裂子树时，不采用拆分恢复的方式，即按子树原有拓扑结构进行恢复，会导致算法的性能下降。但相比 UBA 算法，MPOT 算法的性能下降并不明显，这是由于采用 MPOT 算法构建的组播树的子树也具有较优的子结构。

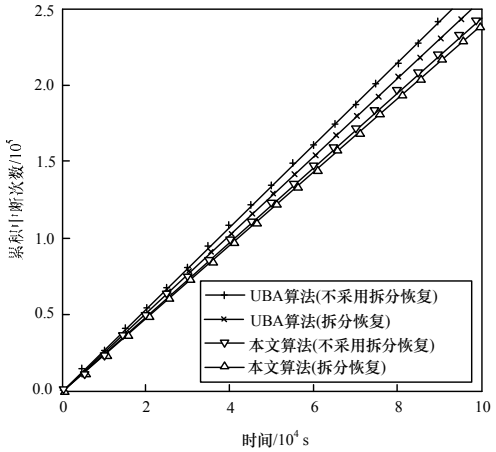

图 4　断裂子树恢复的累计中断次数对比曲线

4　结束语

本文提出了一种基于节点在线时间期望的算法，并针对断裂恢复问题提出改进算法。该算法通过计算节点在线概率以及期望的数学方法作为依据，相比其他基于节点在线时间的算法，不仅得到了具有更高稳定性的组播树，而且考虑了对组播树中断裂子树进行恢复，具有较少的中断次数、更高的稳定性，这更符合实际应用的需求。本文实验均参照实际应用中的情形设定假定条件，由于算法需要遍历组播树，并有大量数值计算，如何对算法进行进一步优化，提高算法效率，并考虑实时传输条件下的时延约束是下一步研究工作。

参考文献

[1] 章　淼，徐明伟，吴建平. 应用层组播研究综述[J]. 电子学报，2004, 32(12): 22-25.

[2] Guo Meng, Ammar M. Scalable Live Video Streaming to Cooperative Clients Using Time Shifting and Video Patching[C]//Proc. of IEEE INFOCOM'04. New York, USA: IEEE Society Press, 2004: 1501-1511.

[3] Sripanidkulchai K, Ganjam A, Maggs B, et al. The Feasibility of Supporting Large-scale Live Streaming Applications with Dynamic Application End-points[C]//Proc. of ACM SIGCOMM'04. New York, USA: ACM Press, 2004: 107-120.

[4] Eveline V, Virgilio A, Wagner M, et al. A Hierarchical Characterization of a Live Streaming Media Workload[C]//Proc. of ACM SIGCOMM Workshop on Internet Measurement. New York, USA: ACM Press, 2002: 117-130.

[5] 罗建光，赵　黎，杨士强. 基于用户行为分析的应用层组播树生成算法[J]. 计算机研究与发展, 2006, 43(9): 1557-1563.

[6] 曾　彬，张大方，黎文伟，等. 基于节点性能估算的应用层组播算法[J]. 计算机工程, 2009, 35(8): 13-16.

编辑　金胡考

(上接第 84 页)

2 种算法针对 Forest Cover Type 数据集的实验结果。其中，除了在第 800 个时间单位 EHPStream 的聚类精度比 HPStream 算法的低外，在其他时间戳上聚类精度都比 HPStream 算法高，基本保持在 80%以上，在第 3 200 个时间单位达到了 91%。

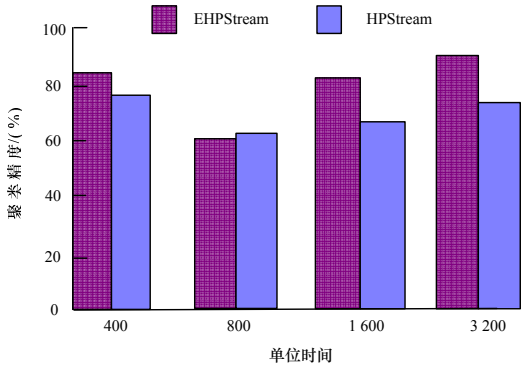

图 3　针对 Forest Cover Type 数据集的聚类精度比较

在算法效率上，由于 EHPStream 算法增加了对类别属性的处理，因此性能有所降低，但是能够满足数据流对算法的要求，几十万条入侵检测数据集在几分钟内就可以处理完。

6　结束语

本文提出一种基于信息熵降维的混合属性数据流聚类算法 EHPStream。通过实验证明，由于 EHPStream 算法增加了对类别属性的处理以及降维的处理，因此在很大程度上提高了聚类精度，在很多时刻的聚类精度有 10%以上的提高。在性能上也能够满足数据流环境的要求。因此，在现实环境中需要处理混合属性的数据流时，宜采用本文的 EHPStream 算法。同时，在实验过程中发现算法不太稳定，在某些时刻聚类精度降低很快，通过对数据集的分析得出，这是因为出现了较大的数据概念漂移。因此，对概念漂移的检测是下一步要开展的工作。

参考文献

[1] Agarwal C C, Han Jiawei, Wang Jianyong, et al. A Framework for Clustering Evolving Data Streams[C]//Proceedings of the 29th International Conference on Very Large Data Bases. Berlin, Germany: [s. n.], 2003: 81-92.

[2] Agarwal C C, Han Jiawei, Yu P S. A Framework for Projected Clustering of High Dimensional Data Streams[C]//Proceedings of the 30th International Conference on Very Large Data Bases. Toronto, Canada: [s. n.], 2004: 852-863.

[3] 姚文集，高明霞. 基于滑动窗口的 XML 数据流聚类方法[J]. 计算机工程, 2010, 36(13): 87-89, 92.

[4] 杨春宇，周　杰. 一种混合属性数据流聚类算法[J]. 计算机学报, 2007, 30(8): 1364-1371.

[5] Huang Zhexue. Extensions to the K-means Algorithm for Clustering Large Data Sets with Categorical Values[J]. Data Mining and Knowledge Discovery, 1998, 2(3): 283-304.

编辑　张　帆