

## 应用层组播的时延受限高稳定性生成树算法\*

曹继军<sup>+</sup>, 苏金树

(国防科学技术大学 计算机学院, 湖南 长沙 410073)

### Delay-Bounded and High Stability Spanning Tree Algorithm for Application Layer Multicast

CAO Ji-Jun<sup>+</sup>, SU Jin-Shu

(School of Computer, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: E-mail: caojijun@nudt.edu.cn

**Cao JJ, Su JS. Delay-Bounded and high stability spanning tree algorithm for application layer multicast. *Journal of Software*, 2010,21(12):3151–3164. <http://www.jos.org.cn/1000-9825/3650.htm>**

**Abstract:** In application layer multicast (ALM) tree, when a parent node quits or fails, all its descendent nodes must adjust their positions, which causes the interruptions of multicast connections. This problem is called stability problem of ALM trees, and it may affect the continuity of multicast data transmission and then degrade user experience seriously. This paper first analyzes the stability problem of ALM trees and proposes the instantaneous stability degree model (ISDM). An approach which takes advantage of the statistical properties of members' join-leave behaviors is proposed to estimate the relative leave probability of member nodes in ISDM. Then, real-time transmission is one of the important aspects of ALM and one major objective of the ALM-based real-time transmissions is to determine multicast tree so as to guarantee the lower delay. This paper proposes the DDSD (the degree-and delay-bounded maximum instantaneous stability degree ALM tree) problem based on the ISDM. The DDSD problem is proved to be NP-Hard and an approximation algorithm (i.e. DDSD-H) is proposed to solve it. In addition, three heuristic policies are presented for the algorithm. Finally, the simulation results demonstrate the validity of the proposed algorithms and the comparison in the performance of the three heuristic policies is made.

**Key words:** application layer multicast (ALM); stability; instantaneous stability degree model (ISDM); spanning tree algorithm; heuristic policy

**摘要:** 应用层组播树会因为单个成员节点的退出或失效而被迫调整其他多个成员节点在组播树中的位置,从而导致多个节点的组播连接被迫中断.该问题被称为应用层组播树的稳定性问题,它严重影响用户接收组播数据的连续性.首先分析了应用层组播树的稳定性问题,提出了瞬态稳定度模型(instantaneous stability degree model,简称ISDM).通过利用组播用户动态行为的统计学特性,提出了一种评估该模型中节点相对离开概率的实用方法.其次,由于实时传输是应用层组播技术的主要应用领域之一,进而基于 ISDM 模型提出了延迟受限最大瞬态稳定度组播生成树问题——DDSD(the degree-and delay-bounded maximum instantaneous stability degree ALM tree),并且证明了该问题属于 NP-Hard 问题.为了解决该问题,提出了 DDSD-H 近似算法,该算法共衍生出 3 种启发式策略.最后,通过

\* Supported by the National Natural Science Foundation of China under Grant No.906004006 (国家自然科学基金); the National Basic Research Program of China under Grant No.2009CB320503 (国家重点基础研究发展计划(973))

Received 2008-07-18; Revised 2008-12-09; Accepted 2009-03-31; Published online 2010-02-11

仿真实验分析比较了所提算法在各种启发式策略下的有效性。

关键词: 应用层组播;稳定性;瞬态稳定度模型;生成树算法;启发式策略

中图法分类号: TP393 文献标识码: A

由于技术和市场等原因,IP组播目前并未得到广泛部署和应用<sup>[1,2]</sup>。近年来,备受瞩目的应用层组播技术有效解决了许多与IP组播相关的问题,但同时也面临新的挑战。例如,尽管IP组播和应用层组播通常都以树结构为数据转发路径,但这两种组播树对成员节点动态性(节点的加入、退出或失效)的适应能力差别很大。应用层组播树会因为单个节点的退出或失效而被迫调整其他多个节点在组播树中的位置,从而导致多个节点的组播连接被迫中断。该问题被称为应用层组播树的稳定性问题,它严重影响组播数据传输的连续性。

目前,针对应用层组播树稳定性问题,相关研究已提出多种旨在降低连接中断次数的生成树算法。这些算法主要分为两类:(1) 基于节点带宽的算法<sup>[3-5]</sup>。此类算法的基本思想是,通过构造较矮的组播树以减少受节点退出或失效行为影响的节点;(2) 基于节点在线时间的算法<sup>[3]</sup>。由于应用层组播节点的会话时间长度分布呈现重尾现象(heavy tails)<sup>[3,6,7]</sup>,意味着在线时间较短的节点比在线时间较长的节点更容易发生退出或失效行为。因此,将在线时间较长的节点调整到离根节点较近的位置以减少受节点退出或失效影响的节点,是此类算法的基本思想。由于上述两类算法构造的组播树分别是带宽排序(bandwidth ordered简称BO)树和在线时间排序(time ordered,简称TO)树,因此,文献[8]将这两种算法分别称为BO算法和TO算法。

BO算法和TO算法对应用层组播树稳定性问题的认识较为感性,相关文献并未给出明确的问题模型,因而其性能受到局限。而要定义该问题模型,首先需要给出衡量应用层组播树稳定性的标准。文献[9,10]定义了稳定因子(stability factor)以衡量应用层组播树的稳定性,但由于该定义过于复杂地考虑了影响稳定性的因素(不但考虑节点退出和失效,而且还重点考虑连接拥塞问题),因而难以评估各个时刻节点离开组播树或发生连接拥塞的概率,这也致使难以实际计算出应用层组播树的稳定因子。我们认为,由于采用拥塞控制技术可以降低或消除连接拥塞,因此,由连接拥塞造成的稳定性问题缺乏普遍性,而由节点退出和失效行为造成的稳定性问题更具有普遍性。为此,本文研究关注于由节点退出和失效引起的稳定性问题,并提出了瞬态稳定度模型——ISDM(instantaneous stability degree model),该模型定义了衡量应用层组播树结构稳定性的新标准。

表1列出了现有主要的应用层组播生成树算法的问题模型。由表1可见,目前还未提出以提高应用层组播树稳定性为优化目标的问题模型。实时组播是组播技术的重要应用,所以,如何构造延迟受限且高稳定性的应用层组播树是我们面临的新问题。为此,本文进一步提出基于ISDM的新型问题模型并给出求解算法。

Table 1 Summary of problem models for ALM spanning tree algorithms

表1 应用层组播生成树算法的问题模型总结

Problem model	Problem description
DBM <sup>[11]</sup>	Degree-Bounded minimum spanning tree problem
MDDBST(MDDL) <sup>[12,13]</sup>	Minimum diameter, degree-bounded spanning tree
BDRBST(LDBR) <sup>[12,13]</sup>	Bounded diameter, residual-balanced spanning tree
BDRFBST <sup>[12]</sup>	Bounded diameter, residual fraction-balanced spanning tree
MDM <sup>[14]</sup>	Minimum delay multicast
DCMD <sup>[15]</sup>	Degree-Constrained, minimum delay spanning tree for node-and-edge-weighted directed graph
DCMST <sup>[16]</sup>	Delay-Constrained minimum spanning tree
DCMA <sup>[17]</sup>	Delay-Constrained multicast algorithm
ODSCP <sup>[18]</sup>	Optimal delay constrained shortest path multicast routing

本文第1节提出ISDM模型,并给出了一种评估该模型中节点相对离开概率的方法。第2节基于ISDM模型提出度和延迟受限的最大瞬态稳定度的应用层组播生成树问题模型——DDSD(the degree-and delay-bounded maximum instantaneous stability degree ALM tree),并证明该问题属于NP-Hard问题。第3节提出求解DDSD问题的算法DDSD-H,并给出了应用于该算法的3种启发式策略。第4节通过仿真实验分析比较本文所提算法及其启发式策略的性能。第5节展望下一步研究工作。

## 1 组播树的稳定性

### 1.1 相关定义

节点退出是指节点主动地离开应用层组播会话,节点失效是指在没通告任何其他节点的情况下节点离开应用层组播会话.就后果而言,非叶节点的退出和失效行为都使得该节点离开了组播树,而且都会造成其他一些节点失去与组播树根节点的连通性.为此,本文将节点退出和失效统称为节点离开.

**定义 1.** 对于应用层组播树  $T(V,E)$ ,节点  $v_i(v_i \in V(T))$  离开树  $T$  指的是从树  $T$  中删除节点  $v_i$  和与之相连的边的事件.节点离开事件导致由树  $T$  产生新的图  $T-v_i$ .

**定义 2.** 对于应用层组播树  $T(V,E)$ ,节点  $v_i(v_i \in V(T))$  离开树  $T$  事件发生后,该事件的受影响节点指的是在图  $T-v_i$  中与根节点之间不存在可达路径(path)的点.

显然,节点  $v_j$  是节点  $v_i$  离开树  $T$  事件的受影响节点当且仅当  $v_j$  为以  $v_i$  为根的子树上的节点,即  $v_j$  是  $v_i$  的子孙(descendent)节点.因此,节点  $v_i$  离开树  $T$  事件的所有受影响节点构成的集合表示为  $N_{desc}(T,v_i)$ .

**定义 3.** 对于应用层组播树  $T(V,E)$  而言,节点  $v_i(v_i \in V(T))$  离开树  $T$  事件会导致节点  $v_j(v_j \in V(T))$  成为该事件的受影响节点,则节点  $v_i$  称为节点  $v_j$  的影响源节点.

显然,节点  $v_j(v_j \in V(T))$  是节点  $v_i(v_i \in V(T))$  的影响源节点当且仅当  $v_j$  为从根节点到  $v_i$  的路径上的节点(节点  $v_i$  除外),即  $v_j$  是  $v_i$  的祖先(ancestor)节点.因此,节点  $v_i$  的所有影响源节点构成的集合表示为  $N_{ance}(T,v_i)$ .

由于组播树中各个节点的子孙节点数目不尽相同,因此不同节点离开的受影响节点数目也不尽相同.同时,在不同时刻各个节点离开组播树的可能性也不尽相同.节点离开事件的受影响节点数目(影响范围)不仅与发生离开事件的节点在组播树中的位置有关,而且与组播树自身的结构有关.为了研究组播树结构与节点离开事件影响范围的关系,我们提出了应用层组播树瞬态稳定度模型 ISDM.

### 1.2 瞬态稳定度模型

**定义 4.** 假设应用层组播树  $T(V,E)$  的总节点数目为  $n(T)$ ,  $\Delta n(T)$  表示该组播树中单个节点离开事件的受影响节点数目,并且  $E(\Delta n(T))$  表示  $\Delta n(T)$  的期望值,则应用层组播树的瞬态稳定度(instantaneous stability degree,简称 ISD)定义为  $ISD(T)=1-E(\Delta n(T))/n(T)$ .

瞬态稳定度的概念定义了应用层组播树在某一时刻的稳定性(即瞬态稳定性)的衡量标准.由定义 4 可知,应用层组播树的瞬态稳定性与组播树中节点离开事件的受影响节点数日期望值密切相关.为了计算节点离开事件的受影响节点数日期望值,需要进一步引入节点相对离开概率的概念.

**定义 5.** 假设应用层组播树  $T(V,E)$  的节点集合为  $\{v_1, v_2, \dots, v_{n-1}, v_n\}$ ,  $D$  表示节点离开事件,并且  $l(D)=0$  表示某时刻没有发生节点离开事件,而  $l(D) \neq 0$  表示某时刻发生离开事件.同时,假设  $Set(D)$  表示发生离开事件的节点集合,则本文将节点  $v_i$  的相对离开概率  $p(v_i)$  定义为  $p(v_i)=P\{v_i \in Set(D) | l(D) \neq 0\}$ .

由定义 5 可知,任意节点  $v_i(v_i \in V(T))$  的相对离开概率  $p(v_i)$  为该应用层组播树中发生节点离开事件时,离开节点为该节点的概率.显然,节点相对离开概率为条件概率,它反映了节点离开应用层组播树的相对可能性大小,这区别于数学严格意义上的节点离开概率.对于节点集合为  $\{v_1, v_2, \dots, v_{n-1}, v_n\}$  的应用层组播树  $T(V,E)$ ,假设任意节点  $v_i \in V(T)$  的相对离开概率为  $p(v_i)$ ,其离开事件的受影响节点集合为  $N_{desc}(T,v_i)$ ,则该树的受影响节点数目的期望值  $E(\Delta n(T))$  和瞬态稳定度  $ISD(T)$  的计算如下:

$$\begin{cases} E(\Delta n(T)) = \sum_{i=1}^n p(v_i) \cdot |N_{desc}(T, v_i)| \\ ISD(T) = 1 - \left[ \sum_{i=1}^n p(v_i) \cdot |N_{desc}(T, v_i)| \right] / n \end{cases} \quad (1)$$

**定义 6.** 应用层组播树中节点的受影响概率指发生节点离开事件时该节点为受影响节点的概率.显然,节点的受影响概率等于该节点所有影响源节点的相对离开概率之和.

由定义 6 可知,对于节点集合为  $\{v_1, v_2, \dots, v_{n-1}, v_n\}$  的应用层组播树  $T(V,E)$ ,假设任意节点  $v_i \in V(T)$  的相对离开

概率为 $p(v_i)$ ,其影响源节点集合为 $N_{ance}(T,v_i)$ ,则节点 $v_i$ 的受影响概率 $P_a(v_i)$ 的计算如下:

$$P_a(v_i) = \sum_{v_k \in N_{ance}(T,v_i)} p(v_k) \tag{2}$$

**定理 1.** 应用层组播树的各个节点的受影响概率之和等于该树的受影响节点数目的期望值.

证明:假设应用层组播树 $T(V,E)$ 的节点集合为 $V(T)=\{v_1,v_2,\dots,v_{n-1},v_n\}$ ;假设任意的节点 $v_i \in V(T)$ 的相对离开概率为 $p(v_i)$ ,其影响源节点集合与受影响节点集合分别为 $N_{ance}(T,v_i)$ 和 $N_{desc}(T,v_i)$ ,该节点的受影响的概率为 $P_a(v_i)$ .同时,假设组播树中各个节点的受影响概率之和为 $P_a(T)$ ,则

$$P_a(T) = \sum_{i=1}^n P_a(v_i) = \sum_{i=1}^n \sum_{v_k \in N_{ance}(T,v_i)} p(v_k) = \sum_{i=1}^n p(v_i) \cdot |N_{desc}(T,v_i)| = E(\Delta n(T)) \tag{3}$$

所以,该定理得证. □

### 1.3 相对离开概率的评估

由于受诸多因素影响,虽然用户节点离开组播会话是随机的,但其整体的动态行为却体现出统计学特性.例如,节点的在线时间长度(session-on-time)呈现对数正态分布(lognormal distribution)<sup>[3,6,7]</sup>.该分布的一个显著特征就是重尾现象,即大部分节点的总在线时间较短,而少数节点的总在线时间很长.这也意味着:当节点的在线时间较短时,则它离开组播树的可能性较大;而当节点的在线时间较长时,它离开组播树的可能性较小.该现象是本文利用节点的在线时间评估其相对离开概率的出发点.表 2 列出了本文将要使用的记号和术语.

**Table 2** symbols and terms defined in the paper

表 2 本文定义的一些记号和术语

Symbols	Meanings
$T$	Random variable for the length of multicast session-on-time
$t$	Variable of multicast session-on-time
$t_i$	A time point of multicast session-on-time
$f(x)$	Probability density function (PDF) for lognormal distribution
$F(x)$	Cumulative distribution function (CDF) for lognormal distribution
$LP(t)$	Relative leave probability, when the multicast session-on-time is $t$
$\Delta T$	Update interval, the length of time domains

由于节点的在线时间 $T$ 服从对数正态分布,所以假设其分布参数为 $\mu$ 和 $\sigma$ ,则其在线时间的概率密度函数 $f(t)$ 和累积分布函数 $F(t)$ 分别见式(4)和式(5)所示,其中, $t$ 为任意正实数.图 1(a)和图 1(b)分别为 $f(t)$ 和 $F(t)$ 的图形,其参数取值为 $\mu=5.19, \sigma=1.44$ <sup>[6]</sup>.

$$f(t) = \frac{1}{t\sigma\sqrt{2\pi}} e^{-\frac{(\ln t - \mu)^2}{2\sigma^2}} \tag{4}$$

$$F(t) = \frac{1}{\sigma\sqrt{2\pi}} \int_0^t \frac{e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}}{x} dx \tag{5}$$

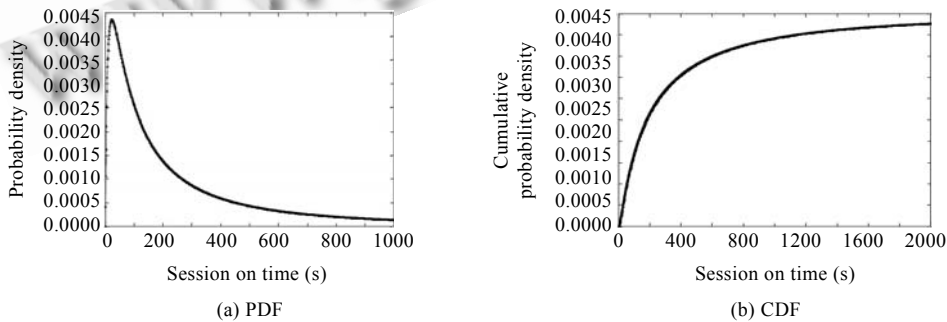


Fig.1 PDF and CDF of lognormal distribution with  $\mu=5.19$  and  $\sigma=1.44$

图 1 对数正态分布( $\mu=5.19, \sigma=1.44$ )的概率密度函数和累积分布函数图

利用节点的在线时间评估其相对离开概率的基本思想是,将时间轴划分为多个区间,如果节点的当前在线时间属于某个区间,则用在线时间长度属于该区间内的概率评估该节点在此刻的相对离开概率.这种评估节点相对离开概率的方法具体描述为:将时间划分为多个长度相等的区间(time domain),即 $[t_0, t_1), [t_1, t_2), [t_2, t_3), \dots, [t_{n-1}, t_n), [t_n, t_{n+1}), \dots$ ,其中 $t_0=0$ ,称区间长度为离开概率更新间隔 $\Delta T$ (update interval).设节点的当前在线时间为 $t$ ,且 $t \in [t_i, t_{i+1})$ ,则此刻该节点的相对离开概率为 $LP(t) = P\{t_i < T \leq t_{i+1}\} = \int_{t_i}^{t_{i+1}} f(x)dx = F(t_{i+1}) - F(t_i)$ .表3列出了8种离开概率更新间隔设置下10个时间区间的相对离开概率评估结果.

**Table 3** Estimated relative leave probability of different update intervals and time domains

**表3** 不同离开概率更新间隔和时间区间下节点相对离开概率评估结果

Update interval (s)	Relative leave probability ( $\cdot 10^{-4}$ )									
	$[t_0, t_1)$	$[t_1, t_2)$	$[t_2, t_3)$	$[t_3, t_4)$	$[t_4, t_5)$	$[t_5, t_6)$	$[t_6, t_7)$	$[t_7, t_8)$	$[t_8, t_9)$	$[t_9, t_{10})$
120	3 899	1 901	1 056	672	463	336	254	198	158	128
240	5 800	1 728	799	452	286	194	139	103	79	62
360	6 856	1 471	610	322	194	127	87	63	47	36
480	7 528	1 251	480	242	140	89	60	43	31	23
600	7 990	1 075	388	188	106	66	44	30	22	16
900	8 686	767	248	112	60	36	23	16	11	8
1 200	9 065	576	172	74	38	22	14	9	6	5
1 800	9 453	360	96	39	19	11	6	4	3	2

已知节点离开概率更新间隔为 $\Delta T$ ,则节点在线时间为 $t$ 时刻的相对离开概率 $LP(t)$ 的计算如下:

$$\begin{cases} t' = t - t \bmod \Delta T \\ LP(t) = F(t' + \Delta T) - F(t') \end{cases} \quad (6)$$

上述方法仅仅是对节点相对离开概率的近似评估,该近似相对离开概率与ISDM模型中的相对离开概率既有区别又有联系.其区别主要体现在:相对离开概率完全反映的是节点离开可能性的相对大小,而且为了便于计算受影响节点数日期望值,任意时刻系统中所有节点的相对离开概率之和为1;而近似相对离开概率反映的是节点在生命周期中各个时刻的离开可能性相对大小,各个节点的近似相对离开概率只与该节点的在线时间有关,每个节点在各个时间区间的近似相对离开概率之和近似为1.两者之间的联系主要体现在:由于各节点在不同时刻的近似相对离开概率采用了统一的评估方法,因此,其近似相对离开概率的大小关系与其相对离开概率的大小关系具有一致性.即假设在任意时刻,节点 $v_i$ 和 $v_j$ 的在线时间分别为 $t_i$ 和 $t_j$ ,则 $LP(t_i) \leq LP(t_j) \Leftrightarrow p(v_i) \leq p(v_j)$ ;反之亦然.同时,节点的相对离开概率和近似相对离开概率的取值范围都为 $[0, 1]$ .为此,在计算节点受影响节点数日期望值和瞬态稳定度过程中,我们用近似相对离开概率代替相对离开概率.为方便起见,下文直接称节点的近似相对离开概率为相对离开概率.

## 2 问题描述

### 2.1 覆盖网络模型

由于任意两个成员节点之间都可以通过IP单播相互到达,所以可将应用层覆盖网络抽象为完全图 $G(V, E)$ .其中: $V$ 为顶点集,顶点表示参与应用层组播的节点; $E=V \times V$ 是边集,边表示应用层组播节点间的最短单播路径;顶点和边都带有权值,顶点的权值表示节点的处理延迟,边的权值表示节点之间的传输延迟;顶点的最大度表示该节点在组播树中最多能容纳的子节点的数目,它主要取决于主机节点的下行带宽.

应用层组播树为图 $G$ 的生成树,其根节点为组播源节点.节点接收组播数据的延迟(简称接收延迟)为根节点到该节点的路径上各个节点的处理延迟和各段单播路径上的传输延迟之和.应用层组播延迟(简称树延迟)为各个叶节点的接收延迟的最大值.

### 2.2 DDS问题

如果单纯地以最大化瞬态稳定度为目标,则导致算法生成延迟较大的应用层组播树.目前,应用层组播技术的主要应用领域是实时传输.因此,如何构造延迟受限的且高稳定的应用层组播树是我们面临的新问题.考虑到

组播数据传输的实时性和稳定性需求,我们希望构造节点度和组播延迟受限、最稳定的应用层组播树(delay-and degree-bounded maximum instantaneous stability degree multicast tree,简称 DDS D),该问题定义如下:

**定义 7(DDS D问题).** 给定完全无向图 $G(V,E)$ ,对于任意节点 $v \in V, p(v) \in [0,1]$ 表示该节点的相对离开概率, $d_{\max}(v) (d_{\max}(v) \in \mathbb{N}^+)$ 表示该节点的最大度限制, $c(v)$ 表示该节点的处理延迟;对于任意边 $e \in E, c(e)$ 表示该边上的传输延迟; $\Delta (\Delta \in \mathbb{R}^+)$ 为给定的树延迟上界.求满足以下条件的生成树 $T$ :(1) 节点 $s \in V$ 为 $T$ 的根节点,且 $V$ 中所有节点都在树 $T$ 中;(2) 假设任意节点 $v \in V$ 的度为 $deg(v)$ ,则 $deg(v) \leq d_{\max}(v)$ ;(3)  $T$ 的树延迟 $D(T) \leq \Delta T$ ;(4) 对于任意满足条件(1)~条件(3)的树 $T', ISD(T) \geq ISD(T')$ .

### 2.3 问题难度分析

证明已知问题是 NP-Hard 问题的常用方法有两种:一种是将问题规约到已知的 NPC 问题;另一种是证明问题是某个已知的 NP-Hard 问题的特例,即给已知的 NP-Hard 问题的实例附加某些特别的限制使得所导出的限制问题与求证的问题等价.本文采用第 2 种方法证明 DDS D 问题是 NP-Hard 问题.为此,我们首先介绍 NP-Hard 的度受限最小平均延迟问题(degree-constrained minimum average-latency problem,简称 DCMAL).给定完全图 $G(V,E)$ ,寻找图 $G$ 的生成树 $T$ ,使树 $T$ 的根节点为 $r$ ,任意节点满足最大度限制,而且各个节点接收延迟的平均值最小.其中, $r$ 为图 $G$ 中的某个指定节点,图 $G$ 中边 $e$ 的权值 $d(e)$ 表示相应节点间的传输延迟,而任意节点 $v$ 的接收延迟为树 $T$ 中从根节点 $r$ 到该节点的路径上各条边的权值之和.文献[19,20]已经证明该问题是 NP-Hard 问题.

**定理 2.** DDS D 问题是 NP-Hard 问题.

证明:已知 DCMAL 问题是 NP-Hard 问题,所以只要证明 DCMAL 的限制问题与 DDS D 问题等价即可.

由定义 4 可知,DDS D 问题等价于节点度和树延迟受限、受影响节点数日期望值最小的应用层组播生成树问题.而由定理 1 可知,树 $T$ 的受影响节点数日期望值等于该树中各个节点的受影响概率之和,因此,DDS D 问题进一步等价于节点度和树延迟受限、受影响概率之和最小的应用层组播生成树问题.

如果令 DDS D 问题中图 $G$ 的任意边 $e(u,v)$ 的权值 $d(e)=p(u)$ ,其中, $p(u)$ 表示节点 $u$ 的相对离开概率,则任意生成树 $T$ 的各个节点的受影响概率之和等于各个节点的接收延迟之和.因此,上述问题可以进一步等价于度受限、接收延迟之和最小的应用层组播生成树问题,所以也等价于度受限最小平均延迟生成树问题.可见,接收延迟受限的 DCMAL 问题等价于 DDS D 问题.定理得证.  $\square$

## 3 DDS D-H 算法

求解 DDS D 问题的启发式算法 DDS D-H 属于贪心算法,它用增量式方法生成节点度和树直径受限的最大瞬态稳定度组播树.算法初始时,树中只存在根节点;算法运行时,首先根据启发式策略(heuristic policy,简称 HP)每次从还未纳入到树中的节点中选择适当的节点,然后将该节点加入到当前树中的受影响概率尽量小的候选加入位置.重复此过程,直到所有节点都纳入到组播树中或算法失败退出.

### 3.1 算法描述

DDS D-H 算法中使用的基本数据结构和函数的定义如下:

**CPPQ:**候选父节点的优先级队列(priority queue of candidate parents).它记录了组播树中可以纳入新的子节点的节点,队列中的节点按照其子节点位置的受影响概率升序排列.

**Delay[v]:**节点的接收延迟数组.它记录了根节点到组播树中其他各个节点的接收延迟,该延迟等于根节点到该节点的路径上各条边的传输延迟和各个节点的处理延迟之和.

**PTA[v]:**节点的受影响概率(probability of to be affected)数组.它记录了组播树中各个节点的子节点位置的受影响概率,其概率值为根节点到该节点的路径上各个节点(包括本节点和根节点)相对离开概率之和.

**Degree[v]:**节点度(degree)数组.它记录了各个候选父节点已经纳入的子节点的数目.

**JNPQ:**待加入节点的优先级队列(priority queue of joining nodes).它记录了未纳入到组播树中的节点,队列中的节点按照节点的相关属性值进行优先级排序,属性的选取方式体现了启发式策略的不同(详见第 3.2 节).

$CTQ, JTQ, TQ$ :临时队列(temporary queue)变量.

$ExtractHead(Q)$ :取去并移走队列  $Q$  中的首个元素,队列中其余元素整体前移 1 位.

DDSD-H 算法的详细描述如图 2 所示.

#### DDSD-H Algorithm

**Input:**  $G(V, E)$

Edge Delay:  $c(u, v)$  for  $u, v \in V$

Node Properties: Delay  $c(v)$ , maximum degree  $d_{\max}(v)$

and Relative Leave Probability  $p(v)$  for any  $v \in V$

Root Node:  $s \in V$  (suppose that  $d_{\max}(v) \geq 1$ )

Up-Bound for Delay of Multicast Tree:  $\Delta$

Heuristic Policy: **HP**

**Output:**  $T(V, E')$ .

**Procedure:**

(1)  $T = \emptyset; CPPQ = \emptyset$ .

(2)  $V(T) \leftarrow s; Degree[] = 0$ .

(3)  $JNPQ \leftarrow V(G) \setminus \{s\}$ , sort the nodes of  $JNPQ$  by **HP**.

(4)  $CPPQ \leftarrow s, Delay[s] = 0, PTA[s] = p(s)$ .

(5)  $CTQ = CPPQ, JTQ = JNPQ, TQ = \emptyset$ .

(6) **while**  $JNPQ \neq TQ$  **do** (7)~(24)

(7)  $TQ = JNPQ$ .

(8)  $CTQ = CPPQ$ .

(9) **while**  $CTQ \neq \emptyset$  **do** (10)~(24)

(10)  $v \leftarrow ExtractHead(CTQ)$ .

(11)  $JTQ = JNPQ$ .

(12) **while**  $JTQ \neq \emptyset$  **do** (13)~(24)

(13)  $u \leftarrow ExtractHead(JTQ)$ .

(14) **if**  $Delay[u] + c(u, v) \leq \Delta$  **then** (15)~(24)

(15)  $V(T) \leftarrow v$ .

(16)  $E' \leftarrow e(u, v)$ .

(17) **if**  $d_{\max}(v) \geq 1$  **then** (18)~(21)

(18)  $JNPQ \rightarrow v$ . //remove  $v$  from  $JNPQ$  and move forward the nodes behind it

(19)  $Delay[v] = Delay[u] + c(u, v) + c(u)$ .

(20)  $PTA[v] = PTA[u] + p(v)$ .

(21)  $CPPQ \leftarrow v$ . //insert  $v$  into  $CPPQ$  by non-decreasing  $PTA$

(22)  $Degree[u] = Degree[u] + 1$ .

(23) **if**  $d_{\max}(u) - Degree[u] = 0$  **then** (24)

(24)  $CPPQ \rightarrow u$ . //remove  $u$  from  $CPPQ$  and move forward the nodes behind it

(25) **if**  $JNPQ = \emptyset$  **then**

(26) **return**  $T(V, E')$ .

(27) **else return** FAILURE.

Fig.2 Pseudo-Code of DDSD-H algorithm

图 2 DDSD-H 算法的伪码表示

算法的执行步骤是:(1) 变量初始化{第 1~5 行}.首先将数据源节点  $s$  纳入空组播树和候选父节点队列  $CPPQ$  中,将其余节点纳入到加入节点的优先级队列  $JNPQ$  中,设置节点  $s$  的接收延迟和子节点位置的受影响概率,初始化队列变量;(2) 解空间搜索{第 6~24 行}.算法每次从  $CPPQ$  中优先选择其子节点位置受影响概率小的节点作为父节点,从  $JNPQ$  中优先选择靠近队列头部的节点纳入树中作为候选子节点,如果该子节点满足最大延迟限制则将节点纳入组播树中{第 15~24 行}.如果新加入节点的最大度不小于 1,则将其插入到  $CPPQ$  中,并从  $JNPQ$  中删除该节点{第 18~21 行}.如果新加入节点的父节点的剩余度减为 0,则将其从  $CPPQ$  中删除{第 24 行};(3) 算法退出{第 25~27 行}.算法退出有两种情况:一种是所有节点都被纳入到树中,即算法成功;另一种是仍然有部分节点未被纳入到树中,即算法失败.算法失败原因主要有 3 种:(1) 所有节点的最大度限制之和小于  $|V|-1$ ;(2) 树延迟上界  $\Delta < \Delta'$  (假设在不考虑稳定性因素的情况下的最小树延迟为  $\Delta'$ );(3) 算法未获得全局最优解.其中,原因(1)和原因(2)是由问题本身造成的,而原因(3)是贪心算法自身缺陷造成的,它导致搜索局部解空间失败,而且未能获得全局最优解.

### 3.2 启发式策略

启发式策略 HP 定义了算法对  $JNPQ$  队列中的节点的排序方式.DDSD-H 算法共有 3 种启发式策略,即节点

度优先、节点相对离开概率优先和综合考虑节点度和相对离开概率的启发式.假设当前  $JNPQ$  队列长度为  $l$ ,则这 3 种启发式策略描述如下:

**策略 1(DPH).**  $\forall i, j \in N$ , if  $1 \leq i < j \leq l$ , then  $d_{\max}(JNPQ[i]) \geq d_{\max}(JNPQ[j])$ .

如果将最大度限制较大的节点优先置于离根节点较近的位置,则将获得高度较低的应用层组播树,那么节点离开事件发生后受影响节点数目将较小,从而得到瞬态稳定度较高的树.这便是节点度优先的启发式(degree priority heuristic,简称DPH)的基本思想.在本文算法中,DPH策略对优先级队列  $JNPQ$  中的节点按照最大度限制值降序排列.

**策略 2(LPPH).**  $\forall i, j \in N$ , if  $1 \leq i < j \leq l$ , then  $p(JNPQ[i]) \geq p(JNPQ[j])$ .

通常,组播树中距离根节点较近的节点具有较多的子孙节点.可见,距离根节点较近的节点如果具有较低的相对离开概率,则整个树的受影响节点数日期望值将较低,从而生成瞬态稳定度较高的组播树.这便是节点相对离开概率优先的启发式(relative leave probability priority heuristic,简称LPPH)的基本思想.在本文算法中,LPPH策略对优先级队列  $JNPQ$  中的节点按照相对离开概率值降序排列.

**策略 3(DLPH).**  $\forall i, j \in N$ , if  $1 \leq i < j \leq l$ , then  $DLPR(JNPQ[i]) \geq DLPR(JNPQ[j])$ .

策略 1 和策略 2 分别优先考虑节点度和节点相对离开概率,其缺点是未能综合考虑这两种因素.为此,策略 3 为综合考虑节点度和相对离开概率的启发式(degree and relative leave probability priority heuristic,简称DLPH)策略.DLPH策略以节点度与相对离开概率的比值(ratio of degree and relative leave probability,简称DLPR)作为优先级队列  $JNPQ$  排序的依据,即  $\forall v \in V(G)$ ,  $DLPR(v) = d_{\max}(v)/p(v)$ .在本文算法中,DLPH策略对  $JNPQ$  中的节点按照DLPR值降序排列.

### 3.3 算法复杂性分析

最后分析DDSD-H算法的时空复杂性.执行该算法伪码的第 1~5 行需要常数时间  $O(1)$ ;伪码的第 6~24 行为 3 层While循环.如果算法成功,则外层循环{第 6~24 行}执行的次数是  $|V|-1$ ,中层循环{第 9~24 行}和内层循环{第 12~24 行}在最坏情况下执行的次数都为  $|V|-1$ .如果采用折半插入,则执行内层循环中的伪码的第 21 行需要进行比较操作的次数为  $O(\log_2|V|)$ .因此,执行伪码第 6~24 行需要的时间为  $O(|V|^3 \cdot \log_2|V|)$ ;执行伪码的第 25~27 行退出算法时需要常数时间  $O(1)$ .可见,DDSD-H算法的时间复杂度为  $O(|V|^3 \cdot \log_2|V|)$ .DDSD-H算法的空间代价主要体现在存储完全图边的权值,故其空间复杂度为  $O(|V|^2)$ .

## 4 性能评价

为了比较 DDSD-H 算法与同类算法的性能,我们实现了 DDSD-H 算法、BO 算法和 TO 算法.在实现 BO 算法时,首先将节点按照最大度限制值降序排列,然后依次将各个节点加入组播树中,每次选择的父节点是已在组播树中的深度最小的非饱和节点(即可以容纳子节点).在实现 TO 算法时,首先将节点按照已在线时间长度降序排列,然后依次将各个节点加入组播树中,每次随机地选择已在组播树中的非饱和节点作为父节点.

### 4.1 实验方法

#### 4.1.1 仿真工具

实验工具为面向对象的离散事件仿真器OMNeT++(object modular network testbed in C++)<sup>[21]</sup>.该仿真器是由布达佩斯技术大学设计的一种内核源代码完全开放的仿真工具,可以运行于Linux和Windows等多个平台.OMNeT++主要使用两种语言:NED(network topology description)和C++.前者用来描述网络拓扑,其中可以定义模块、链路和网络,后者执行NED语言定义的模块.

#### 4.1.2 底层网络拓扑

与文献[22]类似,该实验使用GT-ITM拓扑生成器<sup>[23]</sup>产生基于transit-stub模型的物理网络拓扑.该网络由 4 040 个路由器节点构成,共有 4 个transit区域,每个transit区域有 10 个transit节点,每个transit节点上附着 4 个stub区域,每个stub区域又有 25 个stub路由器.域内节点的物理链路连接方式采用Waxman模型,且参数为



$\alpha=0.6, \beta=0.7$ . transit-transit和transit-stub物理链路延迟均匀随机地赋予 1ms~55ms. 每个stub域内的物理链路延迟均匀随机地赋予 1ms~10ms. 参与应用层组播的主机节点随机均匀地连接到stub路由器, 主机与路由器之间的物理链路延迟均匀随机地赋予 1ms~5ms. 在运行DDSD-H算法之前, 首先用Floyd-Warshall算法<sup>[24]</sup>计算任意路由器节点之间的延迟最短路径, 然后获得任意主机节点之间单播最短路径长度的完全无向图 $G(V, E)$ .

#### 4.1.3 节点最大度限制

假设主机节点的最大度限制 $d_{\max}(v)$ 服从整数区间 $[2, 6]$ 内的均匀分布.

#### 4.1.4 节点的生命过程

与文献[11]的实验方法类似, 我们假设节点的加入时刻服从 $[0, T]$ 范围内的均匀分布, 其参数设置为 $T=3600s$ , 而各个节点参与组播会话的生命周期符合对数正态分布, 其参数设置为 $\mu=5.19, \sigma=1.44$ <sup>[6]</sup>. 所以, 当有 $N$ 个节点参与应用层组播时, 单位时间内加入组播树的节点平均数目为 $N/T$ , 且所有用户节点在 $T$ 时间内全部加入. 在总共进行的 4 次实验中, 参与节点数目分别为 $N_1=500, N_2=1000, N_3=1500$  和 $N_4=2000$ . 而在每次实验中, 我们分别选取 $t=1800s, 3600s, 5400s, 7200s, 9000s, 10800s$ 和 $12600s$ 时刻构造应用层组播树.

#### 4.1.5 离开概率更新间隔

本文实验设置 $\Delta T$ 为 360s.

#### 4.1.6 最大树延迟限制

假设度受限、直径最小的组播树的树延迟为 $DT$ , 则实验中最大树延迟限制设置为 $\Delta=\delta DT$  ( $\delta>1$ ). 在求 $DT$ 时, 我们采用DCMD-H算法<sup>[15]</sup>. DCMD是一个同时考虑了边权、节点权值和最大度约束 3 方面要素的最小延迟应用层组播生成树问题模型, 而DCMD-H算法是一种基于最大延迟路径求解DCMD问题的有效算法. 本文做了两种实验: 一种是设置 $\delta=+\infty$ , 即不限制树延迟. 该情况下的DDSD-H退化为求解度受限的最大瞬态稳定度生成树的算法. 该实验为算法在实际应用中合理地设置 $\delta$ 提供重要依据; 另一种是设置 $\delta$ 为能够有效地平衡组播树延迟和最大瞬态稳定度的值, 本文实验设置 $\delta$ 为 1.35.

#### 4.1.7 仿真实验的目的

比较 DDSD-H 算法在不限制树延迟情况下与 BO 算法和 TO 算法构造的应用层组播树在受影响节点数日期望值和瞬态稳定度等方面的性能; 比较 DDSD-H 算法在 3 种启发式策略下与 DCMD-H 算法所构造的组播树延迟方面的性能, 为实际应用中如何通过合理设置 $\delta$ 而权衡应用层组播的树延迟和稳定性提供参考; 比较 DDSD-H 算法在适当地限制树延迟情况下各种启发式策略构造的组播树在受影响节点数日期望值和瞬态稳定度等方面的性能; 验证构造高瞬态稳定度的应用层组播树与提高整个会话期间应用层组播稳定性的定性关系, 同时对各种算法的下应用层组播中断次数进行比较.

## 4.2 实验结果

我们首先比较了在不限制树延迟情况下的 DDSD-H 算法(其 3 种启发式策略下的具体算法的分别表示为 DDSD-DPH, DDSD-LPPH 和 DDSD-LPDH)与 BO 算法、TO 算法、DCMD-H 算法构造的应用层组播树的相关性能. 组播系统中的节点数目随着组播会话时刻变化而变化, 同时也与节点总数目密切相关. 表 4 给出 4 个组播系统在不同会话时刻的节点数目.

**Table 4** Number of nodes in ALM system for different total numbers and session moments

**表 4** 不同节点总数目的组播系统在不同组播会话时刻的节点数目

	1 800s	3 600s	5 400s	7 200s	9 000s	10 800s	12 600s
$N_1=500$	79	109	48	31	23	18	15
$N_2=1000$	159	217	96	63	46	36	29
$N_3=1500$	238	326	144	95	70	55	44
$N_4=2000$	318	436	192	127	93	72	59

我们分别使用上述 4 种算法在表 4 中各个组播会话时刻构造应用层组播树. 为方便起见, 我们称节点总数目为 500, 1 000, 1 500, 2 000 的组播系统的组播会话分别为 Session 1~Session 4. 图 3 和图 4 分别统计了上述各种算法构造的组播树的受影响节点数日期望值和瞬态稳定度. 结果表明, 各种算法构造的组播树的受影响节点数

日期望值和瞬态稳定度都随着该树中节点数目的增多而增大,反之亦然。

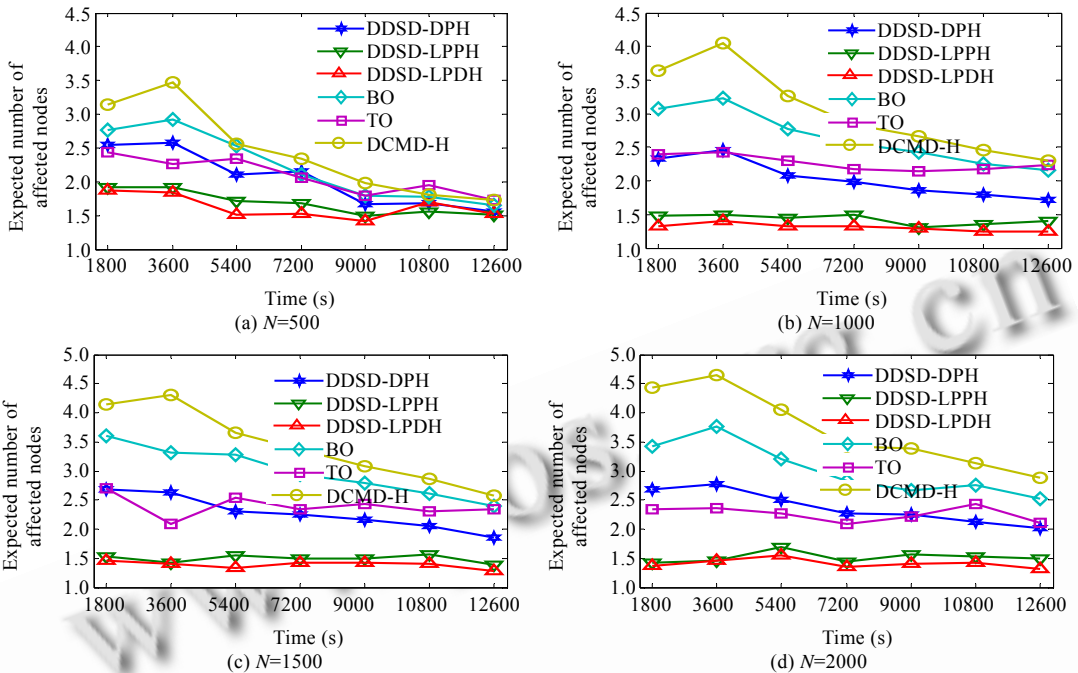


Fig.3 Expected number of affected nodes under different algorithms with  $\delta=+\infty$

图3 各种算法下的受影响节点数日期望值( $\delta=+\infty$ )

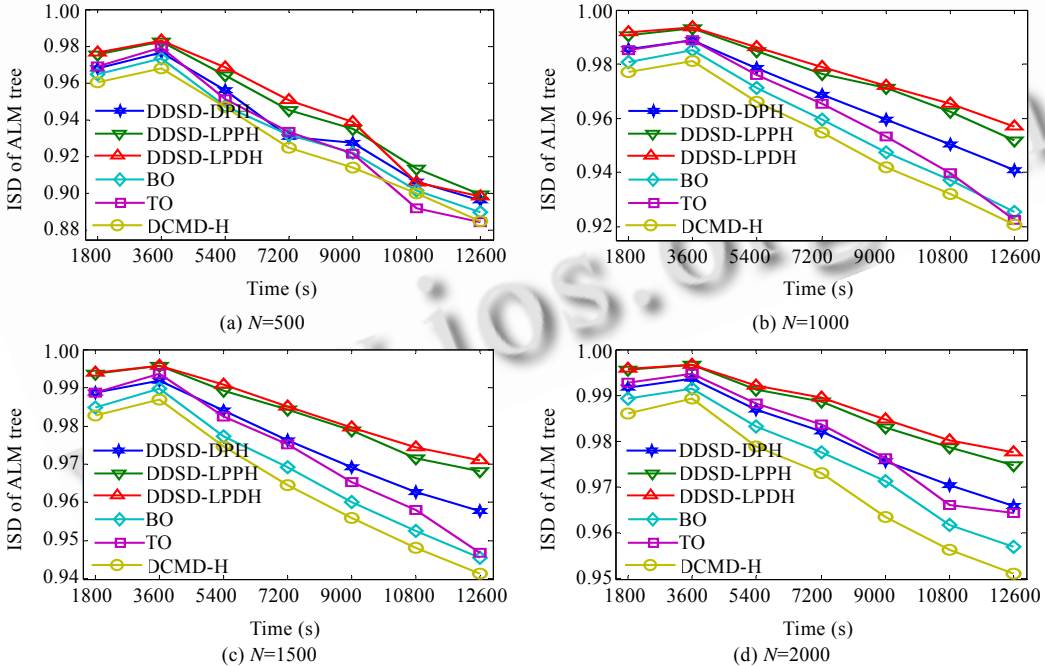


Fig.4 Instantaneous stability degree under different algorithms with  $\delta=+\infty$

图4 各种算法下的瞬态稳定度( $\delta=+\infty$ )

由图3可见:

- (1) 在组播树的受影响节点数日期望值方面,DDS-H 的3种启发式算法优于其他算法,并且 TO 算法优于

BO 算法,DCMD-H 算法由于未考虑稳定性问题,因此它在这方面的性能最差;

(2) 在 DDS-D 的 3 种启发式算法中,DDS-D-LPDH 算法明显优于 DDS-D-DPH,同时,它也略微优于 DDS-D-LPPH 算法.这说明 DDS-D 算法优先选择相对离开概率较小的节点的比优先选择最大度限制较大的节点能够获得更高的瞬态稳定度,而如果能够综合考虑这两种因素,将会取得更好效果;

(3) DDS-D-DPH 和 TO 算法的性能相当.进一步地,当系统中节点数目较多时,DDS-D-DPH 算法略优于 TO 算法;而当系统中节点数目较少时,DDS-D-DPH 算法略差于 TO 算法.

图 5 比较了各种算法构造的应用层组播树在树延迟方面的性能.由该图可见,DCMD-H 算法明显优于其他算法,而其他 5 种算法的性能差异不大.在 DDS-D 的 3 种启发式算法中,DDS-D-LPPH 和 DDS-D-LPDH 算法构造的应用层组播树的树延迟基本相同,它们略优于 DDS-D-DPH 算法.树延迟性能相对最差的是 TO 算法.各种算法在组播树延迟方面的性能差异主要是由于它们各自构造了高度不同的组播树引起的.通常而言,高度较小的组播树的树延迟也较小,反之亦然.同时,根据相关实验数据可知,DDS-D-LPDH 算法构造的应用层组播树的树延迟是 DCMD-H 算法的 1.28 倍~2.09 倍,因此,我们将后续实验中的  $\delta$  设置为 1.35.

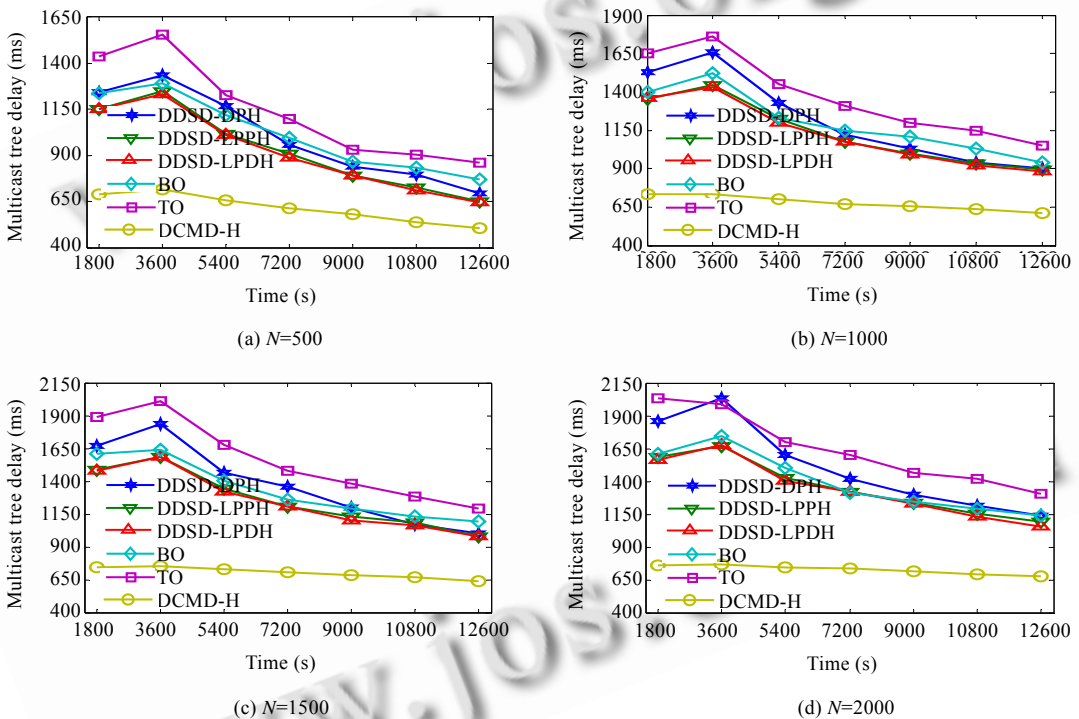


Fig.5 Tree delay under different algorithms with  $\delta=+\infty$

图 5 各种算法下的树延迟( $\delta=+\infty$ )

由于 BO 算法和 TO 算法都没有考虑树延迟限制问题,因此当  $\delta$  设置为 1.35 时,我们只比较了 3 种启发式算法和 DCMD-H 算法的性能.图 6 和图 7 分别比较了各种算法构造的应用层组播树在瞬态稳定性方面的性能.由图 6 和图 7 可见,DDS-D-LPDH 和 DDS-D-LPPH 算法优于 DDS-D-DPH,而 DDS-D-LPDH 算法略微优于 DDS-D-LPPH 算法.在构造高稳定性组播树方面,3 种启发式算法都优于 DCMD-H 算法.综合比较图 6、图 7 和图 3、图 4 可见,由于限制了树延迟,3 种启发式算法在瞬态稳定性方面的性能均有所下降.但当  $\delta=1.35$  时,还未出现算法失败退出的情况.

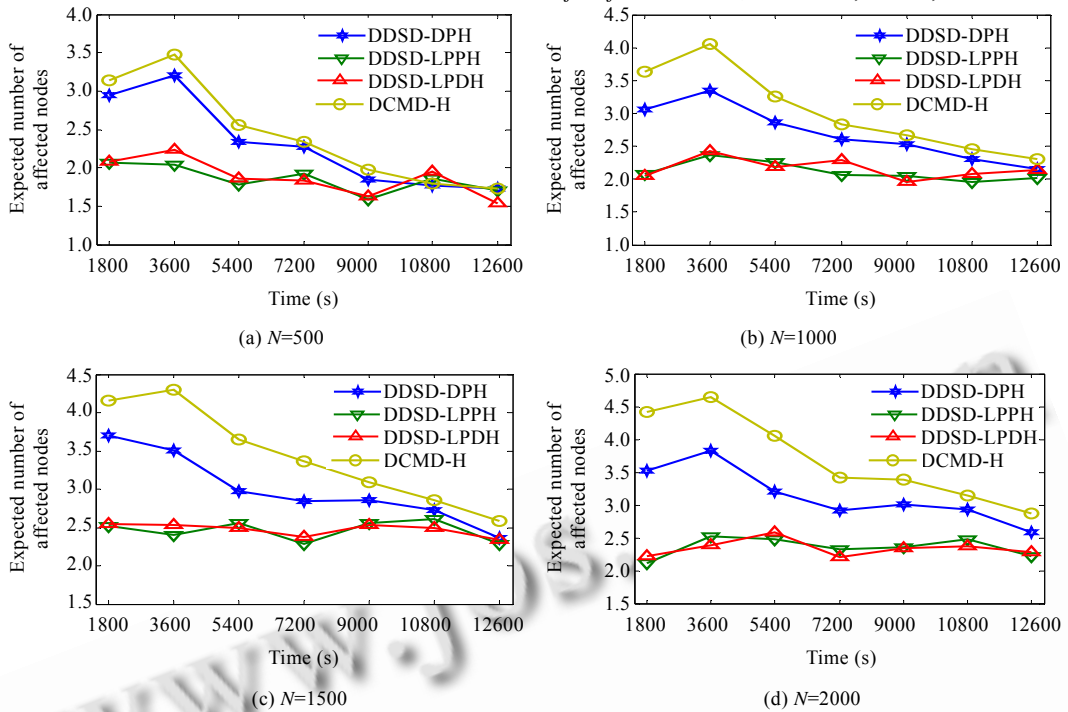


Fig.6 Expected number of affected nodes under different algorithms with  $\delta=1.35$

图 6 各种算法下的受影响节点数期望值( $\delta=1.35$ )

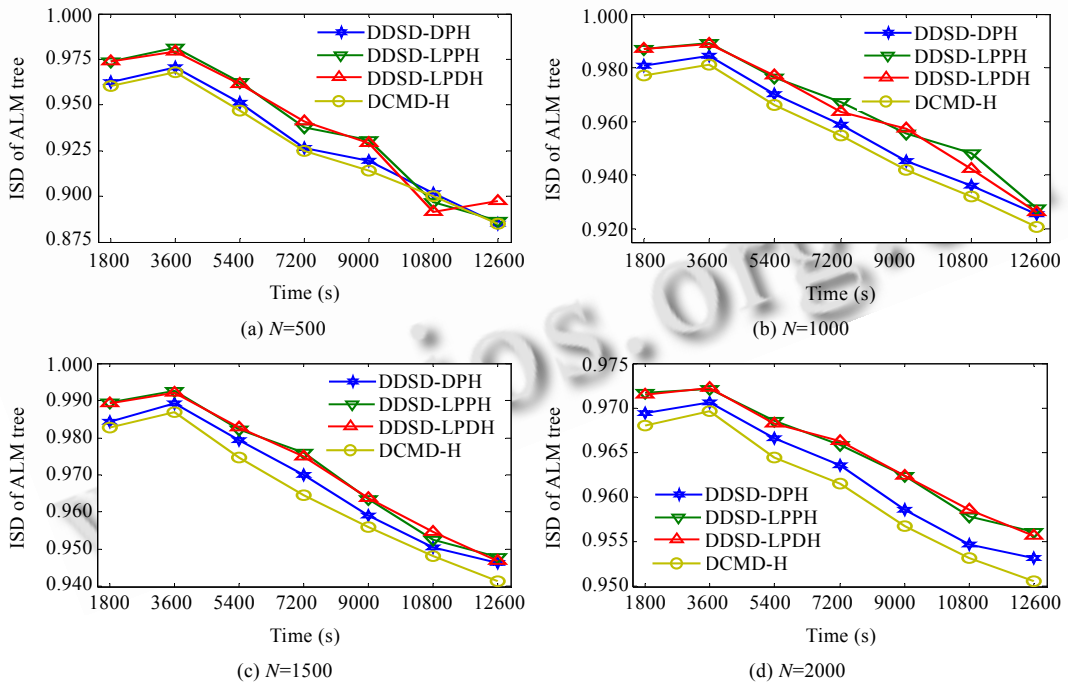


Fig.7 Instantaneous stability degree under different algorithms with  $\delta=1.35$

图 7 各种算法下的瞬态稳定度( $\delta=1.35$ )

由上述实验可知,DDS-H 算法构造的组播树具有较优的瞬态稳定性.为了比较各种算法构造的组播树在整个组播会话过程中的稳定性,我们对各组播会话的累积中断次数进行了统计.为此,假设在整个组播会话过程

中,新节点都随机加入到组播树叶节点位置;同时假设在  $t=1800s, 3600s, 5400s, 7200s, 9000s, 10800s$  和  $12600s$  时刻分别用上述算法计算结果更新应用层组播树的结构;当节点离开时,其子孙节点重新加入到组播树中.在此过程中,我们统计因祖先节点离开而被迫重新加入到组播树的节点的数目(即累积中断次数).图 8 所示结果表明,3 种启发式算法构造的组播树在整个组播会话过程中的累积中断次数明显低于其他算法.例如,与 DCMD-H 算法相比, DDS-D-LPDH 算法将其在 Session 1~Session 4 中的累积中断次数分别降低为 56.84%,36.61%,34.45%和 32.32%.可见,构造和维护具有高瞬态稳定度的应用层组播树对降低整个组播会话过程中累积中断次数有明显效果.

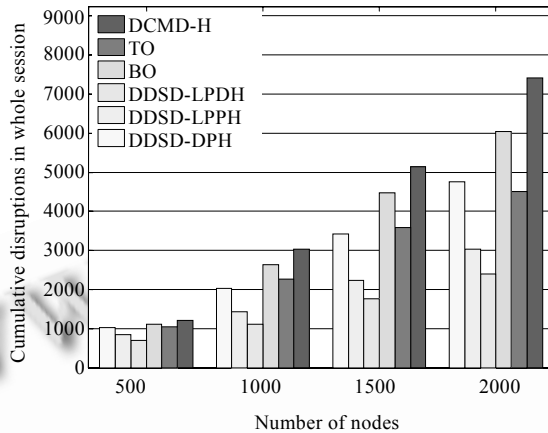


Fig.8 Cumulative disruptions under different algorithms with  $\delta=+\infty$

图 8 各种算法下的累积中断次数( $\delta=+\infty$ )

## 5 下一步工作

针对应用层组播传输的在组播时延和稳定性方面的需求,本文主要研究了如何构造时延受限高稳定性应用层组播树的问题.在未来的研究中,我们将继续完善评估 ISDM 模型中节点相对离开概率的方法,以更加精确和有效地评估应用层组播系统中成员节点退出或失效的相对概率.我们也将进一步探索求解 DDS-D 问题的算法,以获得更优的解法.另外,从分布式和组成员动态管理等方面对本文提出的算法进行改进,使之适应于成员节点规模较大且动态性较强的应用层组播场景,也是我们下一步研究的方向之一.同时,基于本文所提出的模型和算法构建实际的运行于互联网的应用层组播系统,从而进一步验证其有效性,也是我们下一步的研究目标.

## References:

- [1] Deering S. Host extensions for IP multicasting. IETF RFC 1112, 1989. <http://www.ietf.org/rfc/rfc1112.txt>
- [2] Diot C, Levine BN, Lyles B, Balensiefen KH. Deployment issues for the IP multicast service and architecture. IEEE Network, 2000, 14(1):78–88.
- [3] Sripanidkulchai K, Ganjam A, Maggs B, Zhang H. The feasibility of supporting large-scale live streaming applications with dynamic application end-points. In: Proc. of the ACM SIGCOMM. New York: ACM Press, 2004. 127–140. <http://portal.acm.org/citation.cfm?id=1015480>
- [4] Guo M, Ammar M. Scalable live video streaming to cooperative clients using time shifting and video patching. In: Proc. of the IEEE INFOCOM. New York: IEEE Society Press, 2004. 1501–1511. [http://www.ieee-infocom.org/2004/Papers/32\\_3.PDF](http://www.ieee-infocom.org/2004/Papers/32_3.PDF)
- [5] Padmanabhan VN, Wang HJ, Chou PA, Sripanidkulchai K. Distributing streaming media content using cooperative networking. In: Proc. of the ACM NOSSDAV. New York: ACM Press, 2002. 177–186. <http://portal.acm.org/citation.cfm?id=507695>
- [6] Eveline V, Virgilio A, Wagner M, Azer B, Shudong J. A hierarchical characterization of a live streaming media workload. In: Proc. of the ACM SIGCOMM Workshop on Internet Measurement. New York: ACM Press, 2002. 117–130. <http://portal.acm.org/citation.cfm?id=637220>
- [7] Sripanidkulchai K, Maggs B, Zhang H. An analysis of live streaming workloads on the Internet. In: Proc. of the ACM SIGCOMM. New York: ACM Press, 2004. 41–54. <http://portal.acm.org/citation.cfm?id=1028795>

- [8] Tan G, Jarvis SA. Improving the fault resilience of overlay multicast for media streaming. *IEEE Trans. on Parallel and Distributed Systems*, 2007,18(6):721–734.
- [9] Shi F, Wu JP, Xu K. Stability of a multicast tree in cumulative layered multicast congestion control. In: *Proc. of the IEEE Int'l Performance Computing, and Communication Conf. (IPCCC)*. New York: IEEE Society Press, 2003. 725–731. <http://ieeexplore.ieee.org/iel5/8553/27061/01203734.pdf?arnumber=1203734>
- [10] Xu K, Liu JC, Fu LZ, Liu CY. On the stability of application-layer multicast tree. In: *Proc. of the 19th Int'l Symp. on Computer and Information Sciences (ISCIS)*. Springer-Verlag, 2006. 401–412. <http://www.springerlink.com/index/w72q74677797u62.pdf>
- [11] Konemann J, Ravi R. A matter of degree: Improved approximation algorithms for degree-bounded minimum spanning trees. In: *Proc. of the ACM Symp. Theory Computer*. New York: ACM Press, 2000. 537–546. <http://portal.acm.org/citation.cfm?id=586845.586951>
- [12] Shi S, Turner J, Waldvogel M. Dimensioning server access bandwidth and multicast routing in overlay networks. In: *Proc. of the 11th Int'l Workshop on NOSSDAV*. New York: ACM Press, 2001. 83–91. <http://portal.acm.org/citation.cfm?id=378357>
- [13] Shi S, Turner J. Multicast routing and bandwidth dimensioning in overlay networks. *IEEE Journal on Selected Areas in Communications*, 2002,20(8):1444–1455. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1038576](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1038576)
- [14] Broach E, Shavitt Y. Approximation and heuristic algorithms for minimum delay application-layer multicast trees. In: *Proc. of the IEEE INFOCOM*. New York: IEEE Society Press, 2004. 2697–2707. <http://portal.acm.org/citation.cfm?id=1279678>
- [15] 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>
- [16] Salama HF, Reeves DS, Viniotis Y. The delay-constrained minimum spanning tree problem. In: *Proc. of the 2nd IEEE Symp. on Computers and Communications*. New York: IEEE Society Press, 1997. 699–703. <http://portal.acm.org/citation.cfm?id=845348>
- [17] Zhang BX, Krunz M, Chen CJ. A fast delay-constrained multicast routing algorithm. In: *Proc. of the IEEE Int'l Conf. on Communications (ICC 2001)*. New York: IEEE Society Press, 2001. 2676–2680. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=936636](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=936636)
- [18] Mokbel MF, El-Hawet WA, El-Derini MN. A delay-constrained shortest path algorithm for multicast routing in multimedia applications. In: *Proc. of the IEEE Middle East Workshop on Networking*. Lebanon: IEEE Society Press, 1999. 1–6. <http://slog.dk/svn/home/jensen/speciale/trunk/articles/out/Berlut99.pdf>
- [19] Banerjee S, Kommareddy C, Kar K, Bhattacharjee B, Khuller S. Construction of an efficient overlay multicast infrastructure for real-time applications. In: *Proc. of the IEEE INFOCOM*. New York: IEEE Society Press, 2003. 1521–1531. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1208987](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1208987)
- [20] Blum M, Chalasani P, Coppersmith D, Pulleyblank B, Raghavan P, Sudan M. The minimum latency problem. In: *Proc. of the ACM Symp. on Theory of Computing*. New York: ACM Press, 1994. 163–171. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.57.1035>
- [21] OMNeT++: Discrete event simulation system. <http://www.omnetpp.org>
- [22] Fahmy S, Kwon M. Characterizing overlay multicast networks and their costs. *IEEE/ACM Trans. on Networking*, 2007,15(2):373–386.
- [23] Zegura EW, Calvert KL, Bhattacharjee S. How to model and Internetwork. In: *Proc. of the IEEE INFOCOM*. New York: IEEE Society Press, 1996. 594–602. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=493353](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=493353)
- [24] Cormen TH, Leiserson CE, Rivest RL. *Introduction to Algorithm*. 2nd ed., Cambridge: MIT Press, 1990. 558–565.

#### 附中中文参考文献:

- [15] 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>



曹继军(1979—),男,陕西汉中,人,博士生,主要研究领域为高性能路由器技术,互联网组播技术.



苏金树(1962—),男,博士,教授,博士生导师,CCF高级会员,主要研究领域为计算机网络,信息安全.