

高稳定的可扩展覆盖网多播算法

沈晔^{1,2}, 冯径¹, 王占丰³

(1. 解放军理工大学气象海洋学院, 江苏 南京 211101; 2. 95871 部队, 湖南 衡阳 421000; 3. 93615 部队, 天津 300000)

摘要: 提出了一种高稳定的可扩展覆盖网多播 (SOM-HS, scalable overlay multicast with high stability) 算法。SOM-HS 算法定义了节点稳定度因子以及链路权重, 能保证高稳定的节点位于多播树骨干网中。在分层分簇构造过程中, SOM-HS 算法限制节点出度, 保证节点负载均衡。实验结论表明, 与现有其他算法相比, 在不同组规模下, 使用 SOM-HS 算法时的最大多播延时都最小。

关键词: 覆盖网多播; 稳定度因子; 连通率; 分层管理; 节点分簇

中图分类号: TP393.07

文献标识码: A

Scalable overlay multicast algorithm with high stability

SHEN Ye^{1,2}, FENG Jing¹, WANG Zhan-feng³

(1. Institute of Meteorology and Oceanography, PLA University of Science and Technology, Nanjing 211101, China;

2. The 95871 Troop of PLA, Hengyang 421000, China; 3. The 93615 Troop of PLA, Tianjin 300000, China)

Abstract: A new scalable overlay multicast algorithm with high stability (SOM-HS) was proposed. In SOM-HS algorithm, the node stability and the link weight factor were defined to ensure that the high stability nodes were located in the backbone network of multicast tree. In the hierarchical clustering configuration process, SOM-HS algorithm limits the degree of the nodes to ensure load balancing. The experimental results show that, the maximum delay with SOM-HS algorithm is minimum compared to the other existing algorithms in different multicast scales.

Key words: overlay multicast, stability factor, connectivity rate, hierarchical arrangement, node clustering

1 引言

IP 多播需要路由器的支持, 由于网络结构及管理上存在的问题, 使 IP 多播未能在 Internet 上得到广泛的应用^[1]。相比 IP 多播, 覆盖网多播不需要改变已有网络结构, 将多播功能从路由器移到主机上, 随着 Internet 内容服务的增加, 覆盖网多播技术越来越受到重视, 特别是在 P2P 服务模式下的多媒体内容分发, 对于构造稳定、可扩展的覆盖网多播树的算法提出了新的挑战^[2-5]。

在覆盖网多播业务中, 参与多播的节点相互间是对等的, 参与转发的节点通常是“志愿者”, 其以一定概率在线, 而传统的覆盖网多播算法通常假设节点行为是一致的^[5,6], 不适合持续性实时应用场

景 (如视频会议、网络游戏、远程教育和网络电视等), 在这些应用场景中, 节点规模变化较大, 且节点行为不一, 对实时数据的一致性要求高, 不能容忍多播拓扑结构的频繁调整, 因此希望能建立一种转发节点相对稳定的多播拓扑结构。

目前, 提高应用层多播树稳定性主要有以下 3 种措施^[6]: 1) 降低节点离开事件发生的频率; 2) 缩短节点离开后多播树恢复时间; 3) 减小节点离开事件的影响范围。第 1 种措施主要是通过多播网络中部署高性能的专用代理服务器, 用户主机通过专用代理服务器享受多播服务, 降低中转节点的离开频率, 由于部署专用代理服务器需要提前规划代理服务器的物理位置, 需要用户主机相对固定, 可部署性较差。第 2 种措施通过减少多播树断裂后

收稿日期: 2015-07-13; 修回日期: 2016-03-20

基金项目: 国家自然科学基金资助项目 (No.61371119); 国防预研基金资助项目 (No.407010401)

Foundation Items: The National Natural Science Foundation of China (No.61371119), The National Defense Pre-research Foundation of China (No.407010401)

的恢复时间,从而提高多播服务的稳定性,该措施不是从源头上解决多播稳定性问题,而是多播出现不稳定后的补救措施,已有大量学者进行了研究^[7-9],不作为本文研究重点。第 3 种措施分析多播成员节点的稳定性,将稳定节点作为多播树的内部节点,而非稳定节点只作为多播树的叶子节点,从而减少节点离开事件的影响范围,该措施无需额外的基础设施支持,可部署性高,是本文的主要研究内容。

为了减少节点离开事件的影响范围,文献[10]提出了通过已在线时长提高多播稳定性的 SMO (senior member overlay) 算法,分 2 个步骤来实现:首先将已在线时间大于特定阈值的节点组织为一个专门的核心网,其次将非核心网节点作为叶子节点,以核心网节点作为父节点。该方法未考虑节点度以及节点间通信时延,难以适用于大规模多播和时延敏感性多播。

Tan 等^[11]提出的 ROST 算法,综合考虑了节点度和已在线时间。ROST 算法的基本思想是由节点的带宽时间乘积(BTP, bandwidth time product)决定节点在树中的位置,邻居节点通过比较 BTP 值而决定是否交换彼此的位置,通过将 BTP 值大的节点交换到多播树的高层而得到相对稳定的多播树。该方法会导致节点在多播拓扑结构上的动态移动,且没有区分带宽与在线时间对多播稳定性影响的权重。

文献[12]提出了瞬态稳定度 ISDM 模型,该模型利用多播用户动态行为的统计学特性,提出一种评估该模型中节点相对离开概率的方法,进而提出了求解延时受限的最大瞬态稳定度多播树问题(DDSD 问题)的启发式 DLPH 算法,以节点度与相对离开概率的比值作为候选子节点队列的排序依据。然而,根据文献[12]的分析,DDSD 问题会出现求解局部解空间失败,不能获得全局最优解。

基于以上算法的不足,本文提出了高稳定的可扩展多播(SOM-HS, scalable overlay multicast with high stability)算法,能满足大规模、时延敏感性多播的数据分发需求。其基本思想是基于节点的在线时间定义了节点条件稳定度因子,链路权值定义为节点间往返延时与父节点条件稳定度因子的比值,并将链路权值作为新加节点选择父节点的依据。在分层分簇的多播树构造过程中,通过高层节

点的簇内子节点下移一层的方式,保证高层节点的出度增长趋势相对平稳,实现多播各分发节点的负载均衡。

2 SOM-HS 算法模型

节点离开会导致链路断裂以及多播树重构,这样会严重影响到多播数据分发的稳定性。多播节点的离开可以分为 2 类:主动离开和被动离开。主动离开是指节点离开时向其他节点发送通知分组;被动离开是指节点离开时通知其他节点,如节点故障、链路断裂等情况。本文提到的节点离开事件不区分主动离开和被动离开。

首先,定义 Mesh 网络模型 $G = (V, E)$, $v \in V$ 为覆盖网中的节点, $e \in E$ 为覆盖网节点间的链路, $n = |E|$ 为网络 G 中节点数, $T(V', E')$ 表示由数据分发节点 V' 和分发边 E' 构成的多播树, $V' \subseteq V, E' \subseteq E$ 。

定义 1 受影响节点(affected node)集合。假设覆盖网多播树 $T(V', E')$ 中节点 v 发生离开事件,受影响节点集合 $N_{an}(v)$ 表示受 v 离开影响,导致与多播根节点没有路径相连的节点集合,即受影响节点是 $T(V', E')$ 中 v 的子孙节点。

定义 2 源影响节点(affected-source node)集合。假设 v_j 是 v_i 的某一个受影响节点,那么 v_i 就是 v_j 的源影响节点,用 $N_{asn}(v)$ 表示 v 的源影响节点集合,即多播树 $T(V', E')$ 中 v 的祖先节点。

定义 3 节点离线概率(probability of session off time)。假设 $v \in V(T)$, 节点 v 在 t 时间段内持续在线,那么节点 v 在 $t + \Delta t$ 时刻不在线的概率。用 $P(OFF(v, t + \Delta t))$ 表示,如式(1)所示。

$$P(OFF(v, t + \Delta t)) = \int_0^{t+\Delta t} f(x) dx \quad (1)$$

其中, $f(x)$ 为节点离开的概率密度函数,根据文献[13]的统计结论, $f(x)$ 服从参数 $\mu = 5.19$, $s = 1.44$ 的对数常态分布,如式(2)所示。

$$f(t) = \frac{1}{ts\sqrt{2p}} e^{-\frac{(\ln(t)-m)^2}{2s^2}} \quad (2)$$

定义 4 节点在线概率(probability of session on time)。假设 $v \in V(T)$, 且 v 在 t 时间段内持续在线,那么 v 在 $(t, t + \Delta t)$ 时间段内持续在线的概率即为节点在线概率,用 $P(ON(v, t + \Delta t))$ 表示,如式(3)所示。

$$P(ON(v,t + \Delta t)) = 1 - P(OFF(v,t + \Delta t))$$

$$= 1 - \int_0^{t+\Delta t} f(x)dx \quad (3)$$

定义 5 条件离开概率 (conditional leave probability)。假设节点 v 在 t 时间段内持续在线, 那么 v 在 $t + \Delta t$ 时刻离线的概率为条件离开概率, 用 $P_{clp}(v,t)$ 表示, 是典型的条件概率, 用式(4)表示。

$$P_{clp}(v,t) = P(OFF(v,t + \Delta t) | ON(v,t))$$

$$= \frac{P(OFF(v,t + \Delta t), ON(v,t))}{P(ON(v,t))}$$

$$= \frac{P(t < x < t + \Delta t)}{P(t < x)}$$

$$= \frac{\int_t^{t+\Delta t} f(x)dx}{\int_t^{\infty} f(x)dx}$$

$$= \frac{F(t + \Delta t) - F(t)}{1 - F(t)} \quad (4)$$

其中, $F(t)$ 表示节点 t 时间段内在线的概率分布函数。

定义 6 稳定度因子 (stability factor)。由于每个节点 v 在任何时刻 t 只能有一种状态, 即在线或离线, 因此可以用 $sf(v,t)$ 表示 v 在某时刻 t 的稳定度因子, 用式(5)表示。

$$sf(v,t) = 1 - P_{clp}(v,t) = 1 - P(OFF(v,t + \Delta t) | ON(v,t)) \quad (5)$$

定义 7 条件稳定度因子 (conditional stability factor)。在 $(0,t)$ 时间内, 将节点 $v (v \in T)$ 及其影响源节点没有节点离开的概率定义为条件稳定度因子, 用 $csf(v,t)$ 表示, 即 $csf(v,t)$ 是指 $(0,t)$ 时间从根节点到节点 v 的链路一直连通的概率。 $csf(v,t)$ 的计算如式(6)所示。

$$csf(v,t) = \prod_{v' \in N_{asn}(v)} sf(v',t) = \prod_{v' \in N_{asn}(v)} (1 - P_{clp}(v')) \quad (6)$$

定义 8 链路权重 (link weight)。设 $v_i, v_j \in V(T)$, $rtt(v_i, v_j, t)$ 为 t 时刻 v_i 到 v_j 的往返延时, 那么用 $W_{ij}(t)$ 表示链路 e_{ij} 在 t 时刻的链路权重, 由式(7)计算。

$$W_{ij}(t) = \frac{rtt(v_i, v_j, t)}{csf(v_j, t)} = \frac{rtt(v_i, v_j, t)}{\prod_{v' \in N_{asn}(v_j)} (1 - P_{clp}(v', t))} \quad (7)$$

为了计算 $rtt(v_i, v_j, t)$, 可以通过在 v_i 和 v_j 之间建

立简单的 TCP 连接, 即可得到两节点间的往返延时, 此外, 还有很多研究者提出了估算 RTT 的方法^[14-16], 本文采用简单的 TCP 连接估算 RTT。

定义 9 平均链路权重 (average link weight)。假设多播树 $T(V', E')$ 的节点总数为 n , 多播树中节点 v_i 与父节点间的链路权重为 W_i , 多播树的平均链路权重 $E(W_T)$ 用式(8)表示。

$$E(W_T) = \frac{\sum_{i=1}^n W_i}{n} \quad (8)$$

表 1 为本文用到的标记汇总。

标记	说明
$G = (V, E)$	Mesh 网络模型
$T(V, E)$	覆盖层多播树拓扑
r	根节点
v	多播树节点
e	多播树链路
n	多播树节点数
$N_{an}(v)$	节点 v 的受影响节点集合
$N_{asn}(v)$	节点 v 的源影响节点集合
t	持续时间
Δt	节点状态采样间隔
$ON(v, t)$	节点 v 在时间 t 内持续在线事件
$OFF(v, t)$	节点 v 在时间 t 内离开事件
$P_{clp}(v, t)$	节点 v 在时间 t 内的条件离开概率
$P_{clp}(v)$	节点 v 的相对离开概率
$sf(v)$	节点 v 的稳定度因子
$csf(v)$	节点 v 的田间稳定度因子
W_{ij}	链路 e_{ij} 的权重
$E(W_T)$	多播树 T 的平均链路权重

3 SOM-HS 算法步骤

3.1 节点连通率

理论上, 覆盖网络、节点与节点是全连通的, 即任意两节点间是可以直接通信, 按照全连通维持节点邻居状态, 复杂度为 $O(N)$, N 为节点数。

连通率是用来评价两节点是否连通的依据, 类似于 Waxman 模型中的连通率^[13]。两节点的邻居关系表明两节点是连通的。假设 $v_i, v_j \in V(T)$, v_i 与 v_j 间的连通率 P_{ij} 服从指数分布, 如式(9)所

示, 根据 P_{ij} 的大小, 可以得到 v_i 与 v_j 是否连通, 见式(10)。

$$P_{ij} = a e^{-\alpha W_{ij}} \quad (9)$$

$$C_{ij} = \begin{cases} 1, & P_{ij} > b \\ 0, & \text{其他} \end{cases} \quad (10)$$

其中, a 、 b 是介于 0 和 1 之间的常量, a 是边的连通率系数, b 是两节点连通率的阈值, 调整 a 能确保每个节点都能连接到多播树, 调整 b 能有效减少每个节点的邻居表的复杂度。式(10)中, C_{ij} 为 1 表示 v_i 与 v_j 连通, 否则不连通。

3.2 节点分簇算法

在 SOM-HS 算法中, 将连通率作为节点分簇依据。与 NICE^[13] 算法类似, SOM-HS 的簇大小为 $(k, 3k-1)$, k 是整数型常量。图 1 是反映底层物理结构的覆盖网拓扑, 节点间的链路表明 2 个节点是连通的, 链路上的数值为节点间连通率。

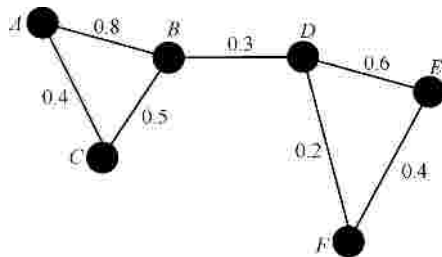


图 1 初始网络拓扑结构

SOM-HS 算法的节点分簇步骤如下。

Step1 得到一个可能的分簇。设簇大小为 3, 随机选择一个节点 (如 B), 可以得到所有包含 B 的可能簇集合为 $\{G'_1, G'_2, G'_3, G'_4\}$, $G'_1 = [ABC]$, $G'_2 = [ABD]$, $G'_3 = [BDE]$, $G'_4 = [BDF]$ 。

Step2 计算各簇的累加连通率。假设 G 是一个簇, 累加连通率 $P_G = \sum P_{ij} (i, j \in G, i \neq j)$ 。因此, 可以得到图 1 中包含 B 的所有可能簇的累加连通率为

$$P_{G'_1} = P_{AB} + P_{AC} + P_{BC} = 0.4 + 0.5 + 0.8 = 1.7$$

$$P_{G'_2} = P_{AB} + P_{BD} + P_{AD} = 0.8 + 0.3 + 0 = 1.1$$

$$P_{G'_3} = P_{BD} + P_{DE} + P_{BE} = 0.3 + 0.6 + 0 = 0.9$$

$$P_{G'_4} = P_{BD} + P_{DF} + P_{BF} = 0.3 + 0.2 + 0 = 0.5$$

Step3 选择最优簇。最优簇 G 应该符合条件 $P_G = \max\{P_{G'_m}\}$, 其中, G'_m 是可能的分簇。由 Step2

结论可知, $\max\{G_1, G_2, G_3, G_4\} = P_{G_1}$, 因此, $G_1 = [ABC]$ 为一个最佳分簇, 并将簇中节点进行标记, 如图 2 所示。

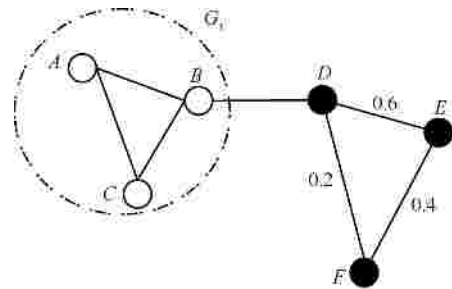


图 2 节点分簇过程

Step4 在没有标记的节点集合中, 继续以上步骤, 直到所有节点都已标记。最后, 每一个节点都是一个簇中的成员, 如图 3 所示。

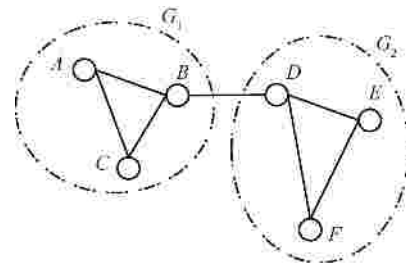


图 3 节点分簇结果

3.3 节点分层管理

节点分层是将节点划分到不同的层上, 层次从底向上排序, 最底层为 L_0 , 如图 4 所示。NICE 算法的分层步骤如下。

Step1 初始化将所有的节点划分到 L_0 层。

Step2 通过分簇算法, 将 $L_i (i > 0)$ 层的节点进行分簇, 并选择簇首节点。

Step3 将 L_i 层的簇首节点加入 L_{i+1} 层, 并将簇首节点移出 L_i 层。

Step4 重复 Step2、Step3 这 2 个步骤, 直到 L_i 层只有 1 个节点, 分层结束。

由 NICE 算法可知, L_i 层的子节点将分别位于 $L_{i-1}, L_{i-2}, \dots, L_0$ 层, 若簇大小为常数 k , 那么第 L_i 层的子节点数为 ik , 随着节点所处层次上升, 其子节点数成倍数上升, 必然导致高层节点的负荷增大。为了保证高层节点的负载均衡, SOM-HS 算法在节点分层过程中进行了改进, 步骤如下。

Step1 初始化将所有的节点划分到 L_0 层。

Step2 通过分簇算法, 将 $L_i (i > 0)$ 层的节点进

行分簇，并选择簇首节点。

Step3 将 L_i 层的簇首节点加入 L_{i+1} 层，并将簇首节点移出 L_i 层。

Step4 当 $i = 1$ 时，选择出簇节点位于 L_{i+1} 层而簇成员位于 L_i 层的簇，将 L_{i+1} 层的簇成员节点进行重新分簇，以链路权重为依据，重新分配到 L_i 层已有的簇中。

Step5 重复 Step2~Step4，直到 L_{i+1} 层只有 1 个节点，分层结束。

相比 NICE 算法的分层过程，SOM-HS 算法多了重新分簇步骤，保证了随节点所处层次的上升，其子节点数目趋于平稳。

SOM-HS 算法的分层分簇框架如图 4 所示，具有以下特点。

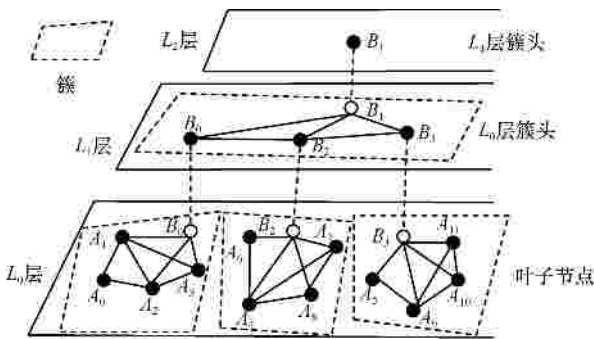


图 4 SOM-HS 算法的节点分层分簇管理

一个节点只能属于某一层的某个分簇。

如果节点在 L_i ($i = 1$) 层，那么该节点肯定是 L_{i-1} 层某分簇的簇首节点。

如果节点属于 L_i ($i = 1$) 层，那么其子节点只属于 L_{i-1} 层，而不会属于其他任何层。

SOM-HS 算法的簇大小为 $O(k)$ ，其中， k 为簇

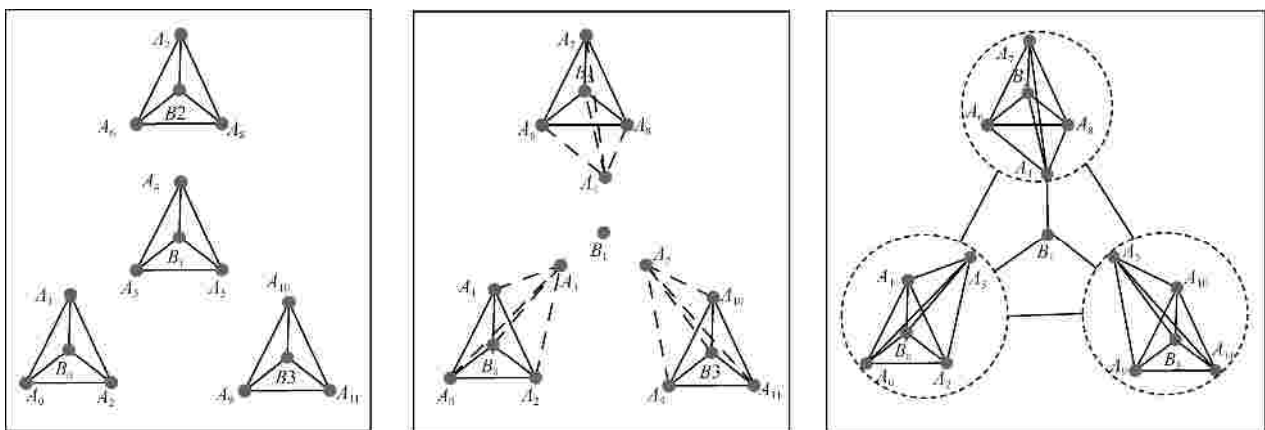
成员数常量。

下面通过具体实例进行说明，如图 5(a)所示，设簇大小为 4，那么 L_0 层分为 4 个簇，分别为 $[A_0, A_1, A_2, B_0]$, $[A_3, A_4, A_5, B_1]$, $[A_6, A_7, A_8, B_2]$, $[A_9, A_{10}, A_{11}, B_3]$ ，簇首节点分别为 B_0, B_1, B_2 和 B_3 。Step2 将 4 个簇首节点从 L_0 层移到 L_1 层，并将其归为同一簇，设 B_1 为簇首。Step2 将 L_1 层的簇首节点 B_1 从 L_1 层移到 L_2 层，此时 B_1 的簇成员节点 $(A_3, A_4, A_5, B_0, B_2, B_3)$ 分布在 2 个层次， $[A_3, A_4, A_5]$ 在 L_0 层， $[B_0, B_2, B_3]$ 在 L_1 层。Step3 将 B_1 在 L_0 层的簇成员 $[A_3, A_4, A_5]$ 重新分簇，划入以 L_1 层成员为簇首节点的分簇中，还是采用链路权重就近原则，假设 A_3, A_4, A_5 分别划入 B_0, B_2, B_3 为簇首的分簇中，最终分簇情形如图 5(b)所示。最终，以 B_0, B_2, B_3 为簇首的簇大小为 4，以 B_1 为簇首的簇大小为 3，如图 5(c)所示。

4 性能测试与分析

为了比较本文提出的 SOM-HS 算法性能，选择 4 种单元多播算法进行比较 NICE^[5]、ROST^[11]、DLPH^[12]、SMO^[10]，其中，NICE 算法是与本文构造框架相似的普适算法，但并没有针对稳定度进行改进，后 3 种算法为优化了稳定度的多播构造算法。

在仿真实验中，本文采用了 NS2 仿真工具，首先通过 GT-ITM 工具生成 10 个 transmission 拓扑，每个 transmission 拓扑包含一个 transport 域和一个 nodes 域。每个拓扑包含 4 个域和 64 个节点 transport 域，每个节点域产生 50 个节点，通过路由器连接到节点。



(a) L_0 层初始分簇结果

(b) B_1 的成员节点重新分簇结果

(c) 最终分层分簇结果

图 5 分层管理步骤

在性能评估中,本文首先对 SOM-HS 算法的时间复杂度进行了分析,然后比较了这 5 种算法的平均链路权重、节点平均中断时间、控制负荷比率以及最大多播延时。

4.1 时间复杂度分析

由第 3 节 SOM-HS 算法步骤可知:计算节点连通率的复杂度为 $O(N)$,节点分簇算法的复杂度为 $O(\log(N))$,节点分层管理重新分簇的复杂度为 $O(k)$,其中, N 为多播节点数, k 为簇大小常数。由于 SOM-HS 算法的以上 3 个步骤是迭代计算的,因此,SOM-HS 算法的时间复杂度为 $O(Nk \log(N))$ 。

4.2 平均链路权重

从式(8)给出的平均链路权重 $E(W_T)$ 计算方法可以得到,平均链路权重是与节点稳定度相关的,因此平均链路权重能表征多播构造树的链路质量,平均链路权重越低,多播构造树越稳定,实时音视频业务的分发效率越高。

图 6 对比了 6 组不同组规模下(组规模分别为 1 000、3 000、5 000、7 000、9 000、11 000),各多播算法平均链路权重 $E(W_T)$ 的受影响程度。从图 6 可以看出,NICE 算法的平均链路权重最高,其余 4 种算法维持了较低的平均链路权重。

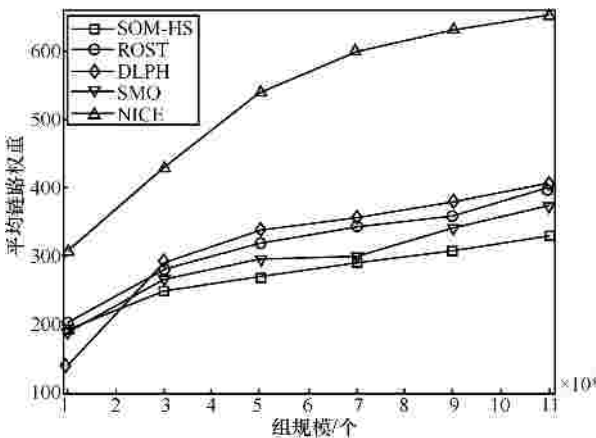


图 6 平均链路压力随组规模的变化关系

为了观察不同时刻 $E(W_T)$ 的受影响程度,设置了 5 000 个多播成员节点,分别在 6 个时刻计算了 $E(W_T)$ 值,如图 7 所示,SOM-HS 的 $E(W_T)$ 要远低于 NICE 算法的 $E(W_T)$ 。综上,图 6 和图 7 表明随着组规模的增大或观察时间的延长, $E(W_T)$ 的上升趋势越稳定,这是因为 SOM-HS 算法中节点维持了最小链路权重的邻居表。在对比的 5 种算法中,特

别是随着组规模的增大,SOM-HS 的平均链路权重具有优势。

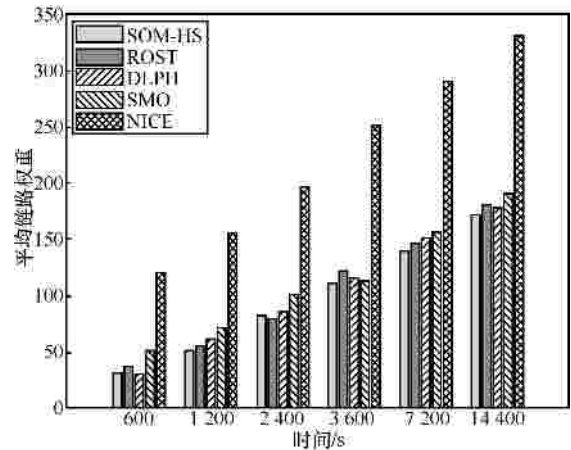


图 7 平均链路压力随时间的变化关系 (N=5 000)

4.3 节点平均中断次数

节点平均中断次数是评价整个数据分发过程中多播树稳定度的典型测度,实验中,节点中断是指该节点的某个祖先节点离开多播树,节点中断不包括节点本身的离开。本实验比较 SOM-HS 算法与 NICE 算法这 2 种分层算法,在连续 10 min 内记录了各多播节点的中断事件。如图 8 和图 9 所示,图 8 是组规模为 5 000 个节点,在 L_2 层(第二高层)和 L_3 层(最高层)的节点比 NICE 算法高,这是由于 SOM-HS 在构造多播树时考虑了节点稳定度,稳定度高的节点被推选到高层中。

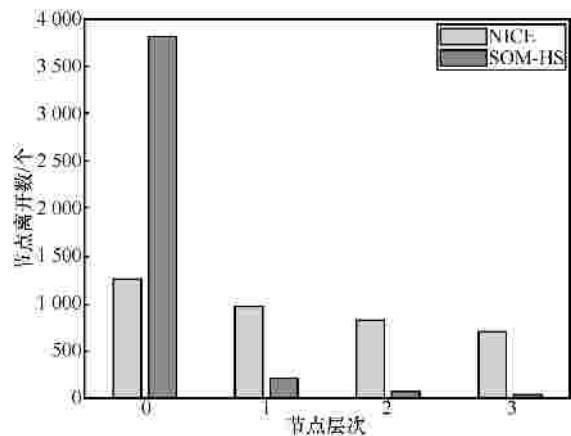


图 8 节点离开数与节点层次关系 (N=5 000)

图 9 是在不同组规模情况下,节点的平均中断次数。由于 SOM-HS 算法中,大多数节点处于 L_0 层,而 L_0 层节点的离开不会导致节点中断,因此 SOM-HS 算法的平均节点中断次数较 NICE 算法更低。

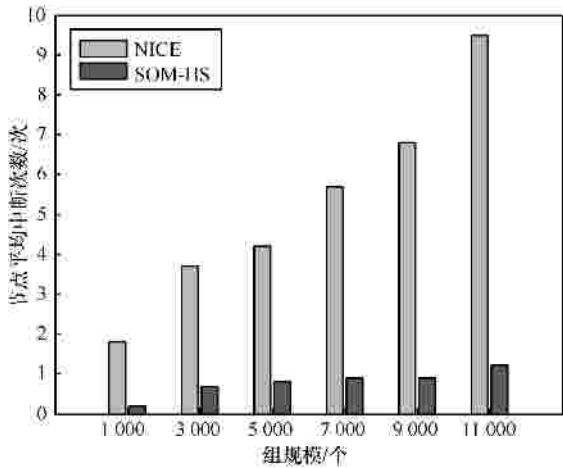


图 9 节点平均中断次数随组规模的变化关系

4.4 控制负荷比率

控制负荷比率代表了链路带宽利用率，实验中，仿真多播源节点分发 50 MB 的文件，分别在 6 组不同组规模情况下。随组规模变化，控制负荷比率的变化趋势如图 10 所示，ROST 算法表现出了较高的控制负荷比率，因为 ROST 算法维持了调整节点位置的 BTP 表，NICE 算法的控制负荷比率较低，主要是因为其不用传输维持多播稳定度的控制信息，SOM-HS 算法与 DLPH 算法、SMO 算法的控制负荷比率较接近，最坏情况为 $O(k)$ ，随着多播规模增大，控制负荷比率增长比较平缓，能适应大规模多播分发需求。

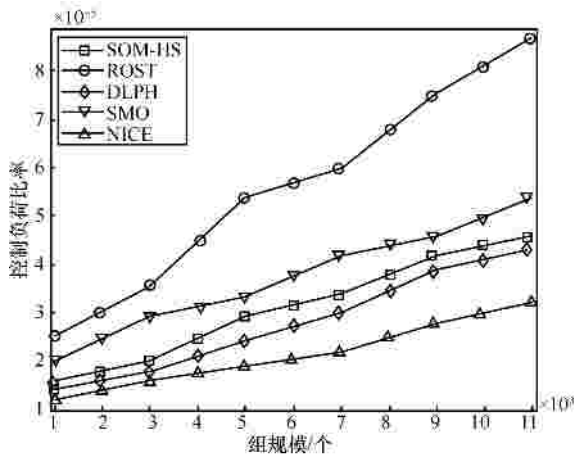


图 10 控制负荷比率随组规模的变化关系

4.5 最大多播延时

最大多播延时是指多播网络中，节点从源节点接收到数据的最大延时。在每个组规模的最大多播延时实验中，多播根节点持续分发 1 000 个数据分组，每个成员节点记录各数据分组的到达时间，先

计算成员节点各数据分组的平均延时，再将成员节点的最大平均延时作为此次组规模的最大多播延时，最大多播延时与组规模的关系如图 11 所示。从图 11 可以看出 SOM-HS 的最大多播延时最小，这是由于在构建分层分簇多播时，链路权重综合考虑了节点稳定度和往返延时，减少了数据的重传时间，SOM-HS 算法多播最大延时为 NICE 算法的 70%，提高显著。

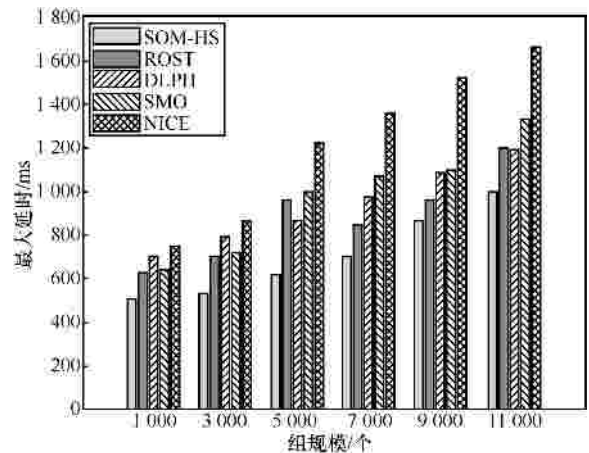


图 11 最大延时随组规模的变化关系

5 结束语

本文提出了一种适合大规模网络的覆盖网多播算法 SOM-HS，在分层分簇的多播树构建基础上，综合考虑了节点稳定度与节点出度，确保构造的多播树相对稳定，同时维持各节点的负载随组规模增大而相对平稳，保证节点负载均衡，理论分析和实验结论表明，相比于已有的多播算法，针对大规模实时数据分发业务，SOM-HS 算法的最大多播延时有显著优势。

然而在实际大规模实时分发应用时，SOM-HS 算法没有考虑节点间的性能差异，下一步将研究节点预测机制，将稳定度高且性能较好的节点靠近多播树的根节点，进一步提高多播分发效率。

参考文献：

[1] 章森, 徐明伟, 吴建平. 应用层多播研究综述[J]. 电子学报, 2004, 32(12A): 22-25.
ZHANG M, XU M W, WU J P. Survey on application layer multicast[J]. Acta Electronica Sinica, 2004, 32(12A): 22-25.

[2] YIN P Y, CHANG R I, CHAO C C, et al. Niche ant colony optimization with colony guides for QoS multicast routing[J]. Journal of Network and Computer Applications, 2014, 40: 61-72.

[3] SANTOS R A, GONZÁLEZ A, VILLASEOR L, et al. Analysis of topological and geographical multicast routing algorithms in wireless ad hoc networks[J]. *Elektronika ir Elektrotechnika*, 2015, 82(2): 23-28.

[4] MCCANNE S. Performing multicast communication in computer networks by using overlay routing: US Patent 8,667,173[P]. 2014-3-4.

[5] CAO J, XIE J, CHEN F. DSD-D: a distributed algorithm for constructing high-stability application-layer multicast tree[C]//FCST. Jilin, China, c2010: 122-128.

[6] 苏金树, 曹继军, 张博锋. 应用层多播稳定性提高技术综述[J]. *计算机学报*, 2009, 32(3): 576-590.
SU J S, CAO J J, ZHANG B F. A survey of the research on ALM stability enhancement[J]. *Chinese Journal of Computers*, 2009, 32(3): 576-590.

[7] YANG M, FEI Z. A proactive approach to reconstructing overlay multicast trees[C]//INFOCOM c2004, Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societ c2004: 2743-2753.

[8] ZONG M, MENG K Y. Restoring delivery tree from node failures in overlay multicast[J]. *IEICE Transactions on Communications*, 2005, 88(5): 2046-2053

[9] KUSUMOTO T, KUNICHIKA Y, KATTO J, et al. Tree-based application layer multicast using proactive route maintenance and its implementation[C]//The ACM Workshop on Advances in Peer-to-Peer Multimedia Streaming. ACM, New York, USA, c2005: 49-58.

[10] TAN G, JARVIS S. Stochastic analysis and improvement of the reliability of DHT-based multicast[C]//INFOCOM 2007, 26th IEEE International Conference on Computer Communications. Anchorage, Alaska, c2007: 2198-2206.

[11] TAN G, JARVIS S. Improving the fault resilience of overlay multicast for media streaming[J]. *IEEE Transactions on Parallel Distributed Systems*, 2007, 18(6): 721-734.

[12] 曹继军, 苏金树. 应用层多播的时延受限高稳定性生成树算法[J]. *软件学报*, 2010, 21(12): 3151-3164.
CAO J J, SU J S. Delay-bounded and high stability spanning tree algorithm for application layer multicast[J]. *Journal of Software*, 2010, 21(12): 3151-3164.

[13] IYER A, KUMAR P, MANN V. Avalanche: data center multicast using software defined networking[C]// 2014 Sixth International Conference on IEEE Communication Systems and Networks (COMSNETS).

Bangalore, c2014: 1-8.

[14] NGUYEN T T, BONNET C. Performance optimization of mult cast content delivery in a mobile environment based on pmipv6[C]// Wireless Communications and Networking Conference (WCNC). Shanghai China, c2013: 1249-1254..

[15] CHITI F, FANTACCI R, SCHOEN F, et al. Optimized random network coding for reliable multicast communications[J]. *mmunications Letters, IEEE*, 2013, 17(8): 1624-1627..

[16] VAN Z E. Layered multicast and fair bandwidth allocation and packet prioritization: U.S. Patent 8,514,718[P]. 2013-8-20.

[17] BANERJEE S, BHATTACHARJEE B, KOMMAREDDY C. Scalable application layer multicast[M]. ACM, 2002.

作者简介：



沈晔 (1985-), 男, 湖南湘潭人, 解放军理工大学博士生, 主要研究方向为覆盖层多播、软件定义网络。



冯径 (1962-), 女, 江苏南京人, 解放军理工大学教授, 主要研究方向为路由调度、软件定义网络。



王占丰 (1982-), 男, 河北临城人, 博士, 93615 部队工程师, 主要研究方向为网络测量、网络大数据分析、下一代网络。