

一种混合的基于分区策略的应用层组播恢复算法

崔建群¹⁾ 陈爱玲¹⁾ 韩 洁²⁾ 常亚楠¹⁾ 吴黎兵³⁾

¹⁾ (华中师范大学计算机学院 武汉 430079)

²⁾ (国家知识产权局专利局专利审查协作湖北中心 武汉 430070)

³⁾ (武汉大学计算机学院 武汉 430079)

摘 要 应用层组播技术由于依靠终端主机转发组播数据,任意中间节点的退出,都将导致其下游节点中断组播数据的接收,因此构建高效的组播恢复算法是提高组播效率的重要措施之一.针对该问题,该文在充分考虑节点性能异构性的基础上,提出了一种混合的基于分区策略的应用层组播恢复算法 HPLR,在该算法中,将节点的服务能力定义为其子孙节点的数目与其根路径长度的比值,再根据节点的服务能力将组播树划分成中心区域和边缘区域,并分别提出了相应的组播恢复算法,以在系统的计算开销和时间开销方面达到平衡.位于中心区域的节点服务能力较强,一旦离开将会造成大面积的节点失联情况的发生.因此,针对中心区域节点失效的情况,该文提出了一种前向式的组播树重构策略 HPLR-CD 算法,在节点失效之前为其孩子节点寻找备份父节点,受影响节点通过与备份父节点联系以恢复数据传输,避免了在节点失联之后仍需计算备份父节点所产生的时间开销,能够快速进行组播树重构,提高组播的效率和性能.位于边缘区域的节点服务能力较弱,针对边缘区域的节点失效情况,该文提出了一种后向式的组播树重构策略 HPLR-MD 算法,当节点失效后,受影响节点通过与祖父节点联系以恢复组播连接,以此避免了为所有节点都计算备份父节点所产生的计算开销,并将节点离开事件对组播树所造成的拓扑结构的变化限制在局部范围.仿真实验表明,HPLR算法的累计百分比在同一重加入时延内均大于对比算法 NICE 以及 BFN,并当累计百分比达到 100%时,HPLR算法平均所需的重加入时延比 BFN 算法小 0.16 s、比 NICE 协议小 0.24 s. HPLR 算法的平均重加入时延分布在 1.05 s 至 1.4 s 之间,随着组播规模的增加其波动范围较小,并且在同一组播规模下其值均小于 NICE 协议和 BFN 算法.此外,HPLR 算法由于采用混合式恢复策略,其在同一规模下维护组播树所需的控制数据包总量均对比算法 Yang 小约 50 KB.

关键词 应用层组播;分区策略;混合的;恢复算法;HPLR

中图法分类号 TP393 **DOI号** 10.11897/SP.J.1016.2018.01990

A Hybrid Partition Based Method for Loss Recovery in Application Layer Multicast

CUI Jian-Qun¹⁾ CHEN Ai-Ling¹⁾ HAN Jie²⁾ CHANG Ya-Nan¹⁾ WU Li-Bing³⁾

¹⁾ (School of Computer Science, Central China Normal University, Wuhan 430079)

²⁾ (Patent Examination Cooperation Hubei Center of the Patent Office, Wuhan 430070)

³⁾ (School of Computer Science, Wuhan University, Wuhan 430079)

Abstract Since application layer multicast (ALM) relies on the terminal hosts to forward multicast data, if the node fails or quits from the multicast tree, its descendants cannot receive the data any more. Thus, it is important to propose an efficient recovery algorithm for the ALM tree to improve the efficiency of data multicast. Aiming at this problem, this paper presents a hybrid partition-based method for ALM tree's recovery, named HPLR (a hybrid partition based method

收稿日期:2016-06-16;在线出版日期:2017-08-19. 本课题得到国家自然科学基金面上项目(61370108,61672257,61170017,61272112)、湖北省自然科学基金(2016CFB145)资助. 崔建群,女,1974年生,博士,教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为网络服务质量、网络管理、高性能计算和应用层组播等. E-mail:jqcui@126.com. 陈爱玲,女,1990年生,硕士,主要研究方向为网络服务质量和应用层组播. 韩 洁,女,1986年生,硕士,主要研究方向为网络服务质量和应用层组播. 常亚楠(通信作者),女,1984年生,博士,讲师,中国计算机学会(CCF)会员,主要研究方向为无线网络、网络管理和应用层组播. E-mail:ynchang@mail.ccnu.edu.cn. 吴黎兵,男,1972年生,博士,教授,博士生导师,主要研究领域为网络通信、网络计算、物联网技术.

for loss recovery), which is based on the heterogeneity of node performance. In this algorithm, the service capability of nodes is defined as the ratio of the number of descendant nodes to the length of the root path. The multicast tree are divided into the center area and the marginal area according to the defined service capacity. Each of the two areas has its own recovery method so as to achieve a balance between computational overhead and time overhead. Since the nodes in the center area have stronger ability for forwarding data, the failure of such nodes will cause interruption for their descendants to receive data. Therefore, aiming at the failure of the center area node, this paper proposes a forward-based multicast tree reconstruction strategy HPLR-CD (a hybrid partition based method for loss recovery- core domain). In this strategy, the backup parent node has already designated for its child nodes before it fails, and the affected node contacts the backup parent node to recover the data transfer. The HPLR-CD algorithm avoids the time cost of the backup parent node when the node is lost, and can quickly reconstruct the multicast tree to improve the efficiency and performance of the multicast. Since the nodes in the margin area has weak ability for forwarding data, aiming at the failure of the margin area node, this paper proposes a backward multicast tree reconstruction strategy HPLR-MD (a hybrid partition based method for loss recovery- margin domain). When the node fails, the affected nodes are in contact with the grandfather node to restore the multicast connection. The HPLR-MD algorithm avoids calculating the computational overhead of the backup parent node for all nodes, and minimize the change of the topology caused by the node failure in ALM tree. The simulation results demonstrate that the cumulative percentage of HPLR algorithm is greater than that of contrast algorithm NICE and BFN in the same rejoin delay. When the cumulative percentage reaches 100%, the rejoin delay of HPLR algorithm is averagely 0.16 s less than that of BFN algorithm, and 0.24 s less than the NICE protocol. The mean rejoin delay distribution of HPLR algorithm is between 1.05 s and 1.4 s, and its fluctuation range is smaller with the increase of multicast size. And the mean rejoin delay of HPLR algorithm is less than the contrast algorithm under the same multicast size. In addition, since HPLR algorithm adopts hybrid multicast recovery strategy, the total amount of control packets required to maintain the multicast tree in the same scale by the HPLR algorithm is about 50 KB less than the Yang algorithm.

Keywords application layer multicast; partition based method; hybrid; loss recovery; HPLR

1 引言

随着通信技术的快速发展,用户对组通信应用的需求与日俱增.组播作为组通信应用中数据分发的核心技术,能有效地增加网络资源的利用率,被广泛地应用于网络视频会议、远程会诊、多方在线游戏、媒体广播等系统中.已有的组播方案分为 IP 组播^[1]和应用层组播^[2-3]两种. IP 组播的路由算法在网络层实现核心功能,由组播中间设备实现数据转发和控制功能,具有较高的数据传输效率和网络带宽利用率,是一种理想的组播方式.但 IP 组播改变了 Internet 基于单播的设计原则,需要改变现有的网络设施,并且难以提供有效的用户认证和计费手

段,缺乏成熟的商业化基础,这些问题致使 IP 组播在全球范围内的部署进展缓慢^[4].鉴于 IP 组播存在的诸多问题,应用层组播于 1997 年由 Francis 提出.应用层组播的特点是将节点组织成一个独立于底层物理链路的逻辑覆盖网络(Overlay Network),在不改变现有的网络设施的基础上,由终端系统代替路由器实现组播数据的复制转发功能,且不需要特殊设备的支持,易于部署,具有较高的灵活性.

应用层组播由于在构成元素方面异于 IP 组播,对端主机的动态性行为表现出不同的反应^[5],致使其存在稳定性方面的问题.应用层组播通常是通过构建组播树来实现数据转发的,通过中间节点的转发使得组播树上的每个节点都能收到信息.当叶子节点退出组播系统时,不会对其他节点产生影响.但

度上提高了应用层组播树的稳定性；其缺点是需要设计重复报文检测策略，以备当节点在接收到重复的报文时能够进行合理的操作，同时转发重复的数据报文消耗了网络的带宽资源，影响了组播树性能的提高，而且并非所有的节点都能接收到除父节点之外的节点转发的额外的报文，因此当该类节点的父节点失效时，PRM 算法对该类节点的数据恢复不发挥作用。

Kusumoto 等人提出了一种为组播树的重构维护备份路由的方法，即 Kusumoto 算法^[18]。Kusumoto 算法的基本思想是：为每个节点预留出度，当发生节点失效的情况时，能够快速地在失效节点的祖父节点和孩子节点之间建立备份路径，从而有效地降低了因节点的动态性行为所导致的额外控制开销。Kusumoto 算法的缺点是它不能有效地利用节点的出度，因此导致重构之后的组播树直径变大，从而导致整个组播树的延迟增大，不利于组播树性能的提高。

文献[23]提出一种基于 gossip 的应用层组播方案，将数据分发过程分为上层分发以及底层簇内分发两个阶段。当节点离开后，其他节点通过 gossip 机制相互协作，共同分担离开节点的转发负载。该方案具有较高的数据分发成功率，但增加了控制开销。

Lei 等人提出了一种类似于软件定义网络(SDN)的应用层组播树恢复策略 DALMLR^[24]。该策略使用一些额外的代理服务器来提供可靠的恢复服务，通过类似于 SDN 的方法计算恢复路径，对恢复数据包进行路由。DALMLR 策略具有恢复速度快以及可靠性高的优势，但需要耗费额外的代理服务器。

文献[25]提出一种分层节点自适应恢复算法 HNA，根据组播树节点性能的不同划分节点的层次，当节点失效时，针对位于不同层次的子节点分别运用设置冗余节点的方法或实时查找父节点的方法进行恢复操作。该算法具有较优的重定向时延，但却未考虑所需的资源开销问题。

文献[26]提出一种基于树结构的树结构的应用层组播模型，利用层次化的树结构来组织、管理节点。当节点退出时，位于不同层次的子节点分别选取合适的父节点重新加入到组播树中。该模型具有较低的传输时延，但没有充分考虑恢复所需的重加入时延以及控制开销等问题。

文献[27]提出的基于备份父节点的故障恢复策略，使被迫中断的节点能够快速重定向，恢复组播服务。

本文针对前向式和后向式恢复算法的优缺点，提出了一种更加全面的策略：基于分区策略的应用

层组播恢复算法，该算法针对组播树中不同区域的节点的性能异构性设计了相应的恢复算法，并通过仿真实验证明了该算法性能的优越性。

3 基于节点服务能力的分区方式

本文在充分考虑节点性能异构性的基础上，提出了一种混合式的基于分区的应用层组播恢复算法 HPLR(A Hybrid Partition Based Method for Loss Recovery)算法。HPLR 算法用节点的子孙节点数量、节点的根路径长度^[28]来描述节点的服务能力。位于组播不同区域的节点其服务能力也不一样，当节点因网络因素导致失效时，其对性能的影响也不相同。服务能力较强的节点一旦离开，将会造成大面积传输中断情况的发生，因此，对于受影响部分节点采用前向式的组播树恢复策略，即在失效节点离开组播树之前就提前为受影响节点选取好备份父节点，避免了在节点失联之后仍需计算备份父节点所产生的时间开销，能够快速进行组播树重构，提高组播的效率和性能；而对于服务能力相对较弱的节点，采用后向式的组播树恢复策略，避免了为所有节点都计算备份父节点所产生的计算开销。

在本章的问题描述中，集合 V 用来表示组播树 T 中的所有节点数目，那么我们有以下定义。

定义 1. 根路径长度。

在组播树中，从根节点 s 到节点 v_i 存在唯一的一条最短路径，在这条最短路径上的节点构成的集合称为根路径。用 $r(v_i)$ 表示节点 v_i 的根路径， $|r(v_i)|$ 表示节点 v_i 的根路径长度，那么

$$|r(v_i)| = \begin{cases} 1, & v_i = s \\ |r(v_p)| + 1, & v_i \neq s \text{ 且 } v_p \text{ 是 } v_i \text{ 的父节点} \end{cases} \quad (1)$$

定义 2. 节点的服务能力。

$desc(v_i)$ 表示节点 v_i 的子孙节点构成的集合， $|desc(v_i)|$ 表示集合中元素的个数， $|r(v_i)|$ 表示节点 v_i 的根路径长度，如式(1)所示。将节点 v_i 的服务能力定义为其子孙节点的数目与其根路径长度的比值，如式(2)所示。

$$reliability(v_i) = \frac{|desc(v_i)|}{|r(v_i)|} \quad (2)$$

对于 $\forall v_i \in V$ ， $|desc(v_i)|$ 的值反映出节点 v_i 含有的下游子孙节点的数量。 $|r(v_i)|$ 的值即节点 v_i 的根路径长度，反映出节点 v_i 与根节点的相对距离。当节点 v_i 的子孙节点的数目 $|desc(v_i)|$ 越大，并且其

根路径长度 $|r(v_i)|$ 的值越小时, 它们之间的比值 $\frac{|desc(v_i)|}{|r(v_i)|}$ 越大, 即节点 v_i 的服务能力 $reliability(v_i)$ 越大, 能够在应用层组播树中承担更多的服务职责。

定义 3. 组播树的中心区域。

对于任意的 $v_i \in V$, n 为组播树中的节点数目, 若 v_i 满足以下不等式

$$reliability(v_i) \geq \frac{1}{n} \sum_{v_i \in V} reliability(v_i),$$

即满足

$$\frac{|desc(v_i)|}{|r(v_i)|} \geq \frac{1}{n} \sum_{v_i \in V} \frac{|desc(v_i)|}{|r(v_i)|}.$$

那么, 将所有符合以上不等式条件的节点构成的区域称为中心区域 (Center Area, CA)。

定义 4. 组播树的边缘区域。

对于任意的 $v_i \in V$, n 为组播树中的节点数目, 若 v_i 满足以下不等式

$$reliability(v_i) \leq \frac{1}{n} \sum_{v_i \in V} reliability(v_i),$$

即满足

$$\frac{|desc(v_i)|}{|r(v_i)|} < \frac{1}{n} \sum_{v_i \in V} \frac{|desc(v_i)|}{|r(v_i)|}.$$

那么, 将所有满足条件的节点构成的区域称为边缘区域 (Marginal Area, MA)。

对于组播树 T 而言, $\frac{1}{n} \sum_{v_i \in V} \frac{|desc(v_i)|}{|r(v_i)|}$ 是所有节点的服务能力的平均值, 对于整个组播树而言是恒定的值。对于 $\forall v_i \in V$, $|r(v_i)|$ 越小表明节点 v_i 的根路径长度越小, 即节点 v_i 越靠近根节点。 $|desc(v_i)|$ 越大则表明由节点 v_i 的子孙节点构成的集合中元素的个数越多, 即其下游的子孙节点数目越多。当节点 v_i 的根路径长度越小, 子孙节点数目越多时, 其节点服务能力 $reliability(v_i)$ 越大。当 $\frac{|desc(v_i)|}{|r(v_i)|} \geq \frac{1}{n} \sum_{v_i \in V} \frac{|desc(v_i)|}{|r(v_i)|}$ 时, 即节点 v_i 的服务能力大于等于组播树中所有节点的服务能力的平均水平, 表示其在组播树中扮演的角色较为重要, 因此, 将此类节点构成中心区域; 反之, 若 $\frac{|desc(v_i)|}{|r(v_i)|} < \frac{1}{n} \sum_{v_i \in V} \frac{|desc(v_i)|}{|r(v_i)|}$, 即节点 v_i 的服务能力未达到组播树中所有节点的服务能力的平均水平, 表示其在组播树中占次要地位, 因此, 由此类节点构成边缘区域。

在任意组播树 T 中, 根节点 s 都应当被划分至中心区域, 而叶子节点都应当被划分至边缘区域。下

面, 将对上述理论进行论证, 以证明公式的正确性。

定理 1. 对于任意的包含 n 个节点的应用层组播树 T , 其根节点 s 总是属于中心区域。

证明. 即需证明 $\frac{|desc(s)|}{|r(s)|} \geq \frac{1}{n} \sum_{v_i \in V} \frac{|desc(v_i)|}{|r(v_i)|}$

恒成立, 由于 $|r(s)| = 1$, 即证 $|desc(s)| \geq \frac{1}{n} \sum_{v_i \in V} \frac{|desc(v_i)|}{|r(v_i)|}$. 对于根节点 s 而言, 其下游子孙节点数目均大于组播树中任何其他节点的子孙节点数目。即 $\forall v_i \in V$, 均有 $|desc(s)| > |desc(v_i)|$, 即 $|desc(s)| > \frac{1}{n} \sum_{v_i \in V} |desc(v_i)| \geq \frac{1}{n} \sum_{v_i \in V} \frac{|desc(v_i)|}{|r(v_i)|}$. 命题得证, 即对于任意的组播树 T , 其根节点 s 总是属于中心区域。证毕。

定理 2. 对于任意的包含 $n(n \geq 2)$ 个节点的应用层组播树 T , 其叶子节点总是属于边缘区域。

证明. 即证对于任意的叶子节点 $v_{leaf} \in V$, 均有

$\frac{|desc(v_{leaf})|}{|r(v_{leaf})|} < \frac{1}{n} \sum_{v_i \in V} \frac{|desc(v_i)|}{|r(v_i)|}$. 由于 $|desc(v_{leaf})| = 0$, 则 $\frac{|desc(v_{leaf})|}{|r(v_{leaf})|} = 0$, 即证 $\frac{1}{n} \sum_{v_i \in V} \frac{|desc(v_i)|}{|r(v_i)|} > 0$. 由于 $\frac{|desc(s)|}{|r(s)|} = n - 1 > 0$, 则 $\frac{1}{n} \sum_{v_i \in V} \frac{|desc(v_i)|}{|r(v_i)|} > 0$ 必定成立, 命题得证, 即对于任意的包含 $n(n \geq 2)$ 个节点的应用层组播树 T , 其叶子节点总是属于边缘区域。证毕。

定理 3 和定理 4 证明, 在任意的 ALM 组播树中, 有且只有一个中心区域和边缘区域, 即组播树仅被划分成中心区域和边缘区域两部分。

定理 3. 在组播树 T 中, 若节点 v_i 属于中心区域, 则其祖先节点同样属于中心区域。

证明. 记 $ance(v_i)$ 为节点 v_i 的祖先节点构成的集合, 由于 v_i 属于中心区域, 则节点 v_i 满足 $\frac{|desc(v_i)|}{|r(v_i)|} \geq \frac{1}{n} \sum_{v_j \in V} \frac{|desc(v_j)|}{|r(v_j)|}$. 对于 $\forall v_j \in ance(v_i)$, v_j 同时满足条件 $|desc(v_j)| > |desc(v_i)|$ 和 $|r(v_j)| < |r(v_i)|$, 则 $\frac{|desc(v_j)|}{|r(v_j)|} > \frac{|desc(v_i)|}{|r(v_i)|} \geq \frac{1}{n} \sum_{v_i \in V} \frac{|desc(v_i)|}{|r(v_i)|}$, 即 $\frac{|desc(v_j)|}{|r(v_j)|} \geq \frac{1}{n} \sum_{v_i \in V} \frac{|desc(v_i)|}{|r(v_i)|}$. 证毕。

定理 4. 在组播树 T 中, 若节点 v_i 属于边缘区域, 则其子孙节点同样属于边缘区域。

证明. 记 $desc(v_i)$ 为节点 v_i 的子孙节点构成的

集合,由于 v_i 属于边缘区域对于任意节点 v_i 满足 $\frac{|desc(v_i)|}{|r(v_i)|} < \frac{1}{n} \sum_{v_j \in V} \frac{|desc(v_i)|}{|r(v_i)|}$. 对于 $\forall v_j \in desc(v_i)$, 节点 v_j 同时满足条件 $|desc(v_j)| < |desc(v_i)|$ 和 $|r(v_j)| > |r(v_i)|$, 则存在不等式 $\frac{|desc(v_j)|}{|r(v_j)|} < \frac{|desc(v_i)|}{|r(v_i)|} < \frac{1}{n} \sum_{v_j \in V} \frac{|desc(v_i)|}{|r(v_i)|}$, 即 $\frac{|desc(v_j)|}{|r(v_j)|} < \frac{1}{n} \sum_{v_j \in V} \frac{|desc(v_i)|}{|r(v_i)|}$. 证毕.

在如图 2 所示的组播树中,根据公式计算树中各个节点的 $reliability(v_i)$ 值,结果如表 1 所示. 根据表 1 的结果,再计算组播树所有节点报务能力的平均值:

$$\begin{aligned} &\frac{1}{n} \sum_{v_i \in V} \frac{|desc(v_i)|}{|r(v_i)|} \\ &= \frac{1}{13} \times \left(\frac{12}{1} + \frac{5}{2} + \frac{5}{2} + \frac{2}{3} + \frac{2}{3} \right) \\ &= \frac{55}{39}. \end{aligned}$$

根据定义 3 和定义 4 对中心区域和边缘区域的判定策略,根节点 s 以及节点 v_1, v_2 属于中心区域,其余节点均属于边缘区域.

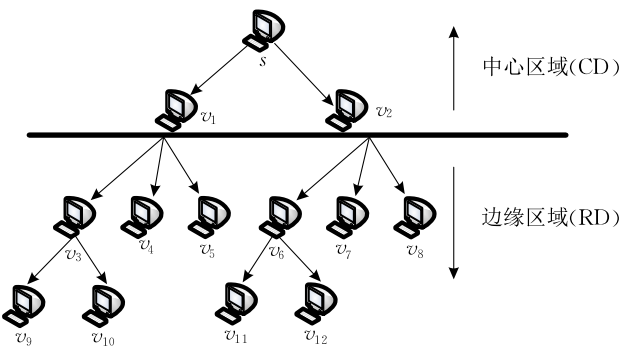


图 2 中心区域与边缘区域的划分

表 1 节点的 $reliability(v_i)$ 值

节点	$\frac{ desc(v_i) }{ r(v_i) }$	是否属于中心区域
s	12	✓
v_1	$5/2$	✓
v_2	$5/2$	✓
v_3	$2/3$	✗
v_4	$0/3$	✗
v_5	$0/3$	✗
v_6	$2/3$	✗
v_7	$0/3$	✗
v_8	$0/3$	✗
v_9	$0/4$	✗
v_{10}	$0/4$	✗
v_{11}	$0/4$	✗
v_{12}	$0/4$	✗

4 基于分区的应用层组播恢复算法

根据节点服务能力的高低划分成为中心区域和边缘区域,针对这两种区域,本节分别提出相应的应用层组播树重构策略. 当位于中心区域的节点失效时,本节提出了一种前向式的组播树重构策略,在节点失效之前为其孩子节点寻找备份父节点以恢复数据传输. 当位于边缘区域的节点失效时,本文提出了一种后向式的组播树重构策略,受影响节点通过与祖父节点联系以恢复组播连接.

4.1 中心区域的节点恢复策略

对于中心区域的节点 v_i ,由于其靠近根节点,且下游节点数目较多,因此,当该区域节点失效时,会对应用层组播造成大面积的影响. 对于这部分节点,应当设计低时延的树重构策略,尽可能快的恢复受影响节点的正常数据接收,从而提高应用层组播的效率.

本小节针对中心区域节点失效的情况提出了一种前向式的组播树重构策略:HPLR-CD(A Hybrid Partition Based Method for Loss Recovery-Core Domain)算法. HPLR-CD 算法采用前向式的组播树恢复策略,能够有效减少节点重定向的时间开销,提升用户体验.

首先将问题模型描述如下:如图 3 所示,假设属于中心区域的节点 v_i 有 n 个孩子节点,其孩子节点构成集合 $desc(v_i) = \{c_0, c_1, \dots, c_{n-1}\}$,从根节点 s 至节点 v_i 的根路径上的祖先节点依次是 $v_0, v_1, \dots, v_{i-1}$,当节点 v_i 失效时,需要为其孩子节点 $c_0, c_1, \dots, c_{n-1}$ 寻找备份父节点,使其能够快速的重定向,恢复数据的传输,且使其满足度约束且代价最小.

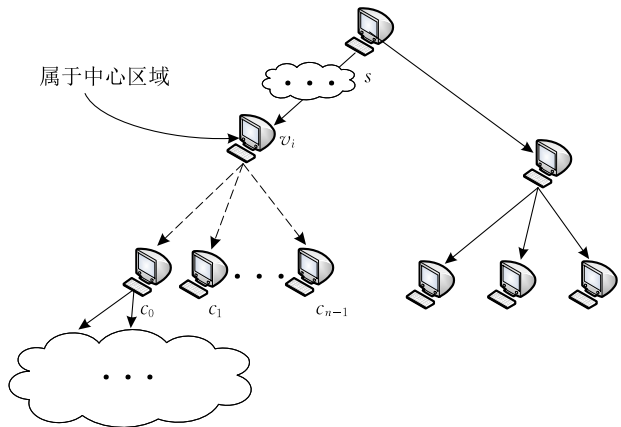


图 3 中心区域的节点 v_i 失效时的问题模型

将覆盖多播的网络模型抽象成完全图 $G(V, E)$, 其中 $v_i \in V$ 是已经加入组播树的节点构成的集合, s 是组播源节点. 设置两个集合 BPS (Backup Parents Set) 和 DSS (Disconnect Subtree Set), 集合 BPS 是备份父节点集合, 其中存放组播树 T 中还有可用出度的节点, 集合 DSS 是失效节点的直接孩子节点构成的集合.

当属于中心区域的节点失效时, 其孩子节点的恢复步骤如下所示:

Step1. 将组播树 T 中存在剩余出度的节点纳入集合 BPS 中;

Step2. 将失效节点的直接孩子节点纳入到集合 DSS 中; 当集合 $DSS \neq \emptyset$ 时, 执行下列步骤:

Step3. 分别从集合 BPS 和集合 DSS 中选择两个节点 v_i, c_j , 使得 v_i 和 c_j 之间的时延开销是最小的一对节点, 则以 c_j 为根节点的子树将以 v_i 为备份父节点;

Step4. 将节点 v_i 的剩余出度减 1, 再次检查 v_i 节点是否还有剩余出度, 若剩余出度为空, 则将 v_i 节点从 BPS 集合中删除;

Step5. 由于节点 c_j 已经找到合适的备份父节点, 因此将该节点从集合 DSS 中移除;

Step6. 若节点 c_j 及其孩子节点集合中存在有剩余出度的节点, 则将该节点加入到集合 BPS 中;

Step7. 重复执行步骤 Step3~6, 直至集合 $DSS = \emptyset$;

HPLR-CD 算法能够在较低的时间开销下快速地恢复组播树, 并且将备用父节点的搜索范围扩大到整个组播树, 即允许任意的非饱和节点作为备用父节点, 充分利用了节点的度, 有效地降低了组播树的直径, 优化了组播树的延迟, 缩短了节点重定向的时间开销.

算法 1. HPLR-CD 算法.

输入: $G(V, E)$;

$cost(e(v_i, v_j))$ for any $v_i, v_j \in V$;

maximum degree $deg_{\max}(v_i)$;

available degree $deg_{avl}(v_i)$;

Initialize $DSS = \{c_i \mid c_i \in children(v_{leave})\}$;

Initialize $BPS = V - \{v_{leave}\} - DSS - \{u \mid deg_{avl}(u) = 0\}$

输出: $map(c_i, p)$, $c_i \in children(v_{leave})$, p is c_i 's backup parent

过程:

1. WHILE $DSS \neq \emptyset$ DO 2~10
2. select $e(c_j, v_i)$, where $c_j \in DSS$, $v_i \in BPS$, such that $cost(e(c_j, v_i))$ is the smallest among those c_j 's and v_i 's;

3. $map \leftarrow (c_j, v_i)$;
4. set v_i as c_j 's backup parent;
5. $degree(v_i) = degree(v_i) - 1$;
6. IF ($deg_{avl}(v_i) = 0$) DO 7
7. $BPS \mapsto v_i$;
8. $DSS \mapsto c_j$;
9. IF $v_i \in (desc(c_j) \cup \{c_j\})$ and $deg_{avl}(v_i) > 0$ DO 10
10. $BPS \leftarrow v_i$
11. END WHILE
12. RETURN map

该算法的主要时间消耗在执行步骤 2 上, 对于步骤 2, 其时间复杂度是 $O(N^2)$, N 表示整个组播树 T 的节点总数, 因此, 该算法的时间复杂度是 $O(N^2)$. 假设组播树中属于中心区域的节点个数为 n_1 ($n_1 \leq N$), 则该算法的空间复杂度为 $O(n_1)$.

4.2 边缘区域的节点恢复策略

本节针对边缘区域的节点失效情况提出了一种后向式的组播树重构策略: HPLR-MD (A Hybrid Partition Based Method for Loss Recovery-Margin Domain) 算法. HPLR-MD 算法采用后向式的组播树恢复策略, 能够有效地避免前向式组播树恢复策略的计算开销.

首先将问题模型描述如下: 如图 4 所示, 假设属于边缘区域的节点 v_i 有 n 个子节点 $\{c_0, c_1, \dots, c_{n-1}\}$, 从根节点 s 至节点 v_i 的根路径上的祖先节点依次是 $v_0, v_1, \dots, v_{i-1}$, 当节点 v_i 失效时, 需要为其孩子节点 $c_0, c_1, \dots, c_{n-1}$ 寻找备份父节点, 使其能够快速的重定向, 恢复数据的传输.

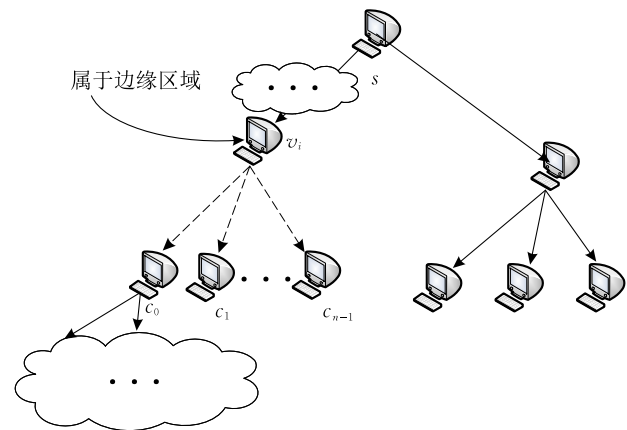


图 4 边缘区域的节点 v_i 失效时的问题模型

属于边缘区域的节点恢复步骤如 Step1~Step3 所示:

Step1. 当节点 v_i 欲离开组播树时, 首先向其父节点 v_{i-1} 发送 $DROP_REQUEST$ 请求消息, 通知父节点自己即将

失效;同时,节点 v_i 向其孩子节点 $children(v_i)$ 发送 $DROP_NOTICE$ 消息,通知孩子节点自己即将离开;

Step2. 节点 v_i 的父节点 v_{i-1} 接收到节点 v_i 发送的 $DROP_REQUEST$ 消息时,首先将断开与节点 v_i 的连接,并将其从自己的孩子节点列表当中删除,同时将 $degree(v_{i-1}) = degree(v_{i-1}) + 1$;

Step3. 节点 v_i 的孩子节点 $children(v_i)$ 接收到 v_i 发送的 $DROP_NOTICE$ 消息后,孩子节点将向其祖父节点 v_{i-1} 发送 $POLL_REQUEST$ 消息,节点 v_{i-1} 将选择在自己度约束范围内的且与自己相距最近的节点进行连接;此时,若还有剩余的节点未能重定向成功,则节点 v_{i-1} 将向自己其它有可用度的孩子节点转发该 $POLL_REQUEST$ 消息,这些节点将首选较大的子树进行连接;若节点 v_{i-1} 不存在有可用度的孩子节点或仍有未重定向成功的节点,则继续向下层节点转发该 $POLL_REQUEST$ 请求消息。

在Step3中,由于节点 v_i 的失效,节点 v_i 的父节点必然存在可用度,因此节点 v_{i-1} 可以在 $children(v_i)$ 中选择与自己相距最近的节点与其建立连接,直至可用度为空,这样的做法可以充分利用节点的出度。其次,对于接收到 $POLL_REQUEST$ 消息的节点,将首选较大的子树进行连接,这样的做法可以有效减小组播树的延时,提高组播效率。

HPLR-MD 算法能够将节点离开事件对组播树所造成的拓扑结构的变化限制在局部范围,因此具有较优的综合性能,是实现组播树快速重构的理想选择。

算法 2. HPLR-MD 算法.

输入: $G(V, E)$

输出: $T(V, E')$

过程:

1. SendMessageToFather($DROP_REQUEST$)


```

{
    disconnect  $v_i$ ;
    remove  $v_i$  from children List;
     $degree(v_{i-1}) = degree(v_{i-1}) + 1$ 
}

```
2. SendMessageToChildren($DROP_NOTICE$);
3. SendMessageToGrandfather($POLL_REQUEST$);
4. SendMessageToChildren($JOIN_RESPONSE$)


```

{
     $k = \min(deg_{avi}(v_{i-1}), n)$ ;
    select  $k$  nodes from  $desc(v_i)$  so these nodes
    have the minimum delay with  $v_{i-1}$ ;
    connect these nodes with  $v_{i-1}$ ;
}

```

remove these nodes from $desc(v_i)$;

IF $desc(v_i) \neq \emptyset$ THEN;

v_{i-1} retransmit $POLL_REQUEST$ to its other children nodes with remnant out degree;

}

HPLR-MD 算法的时间复杂度在最佳情况下是 $O(1)$, 一般情况下, 算法的时间复杂度是 $O(\log N)$, 其中 N 指组播树 T 的节点总数. 假设组播树中属于中心区域的节点个数为 n_2 ($n_2 \leq N$), 则该算法的空间复杂度为 $O(n_2)$.

组播树中位于中心区域和边缘区域的节点的相关信息均存放在集合中, HPLR 算法的空间复杂度为 $O(N)$, 其中 N 为组播树的节点总数, $N = n_1 + n_2$.

5 仿真实验

本节将通过仿真实验对比 HPLR 算法与 NICE 算法和 BFN 算法在平均重加入时延、累计百分比/重加入时延这两方面的性能, 以及 HPLR 算法与 Yang 算法在恢复后组播树的控制开销方面的性能, NICE^[9] 协议是一种典型的分层分簇的应用层组播协议, 它可以支持大量不同的数据转发树, 有较强的可扩展性. BFN 算法^[27] 是一种基于备份父节点的故障恢复策略, 通过父节点备份机制达到快速重定向的目的. Yang 算法^[17] 是一种前向式应用层组播树恢复算法, 当非叶子节点离开时, 其子节点可以快速的加入到备用父节点上。

5.1 仿真实验环境

本文采用的仿真工具为基于 OMNet++^[29] 网络仿真软件的 OverSim 覆盖网络仿真框架. OMNet++ 是一种基于离散事件的、开源的多协议仿真器, 是一个通用的离散仿真框架. OMNet++ 通过 INET 进行网络建模, 将若干实现了基本协议的简单模块组合成为复合模块, 以此成为仿真节点. OverSim 为工作在 OMNet++ 网络仿真环境下的仿真框架, 为结构化的覆盖网络提供通用的功能调用. 该仿真器可以运行于 Linux 和 Windows 等多个平台。

图 5 是当节点数目为 120 时, HPLR 算法的仿真实验图. Overlay Terminal 代表终端节点, 其中 Overlay Terminal[0] 为应用层组播树的根节点. 带箭头的实线代表应用层组播节点之间的邻接关系, 箭头由父节点指向孩子节点。

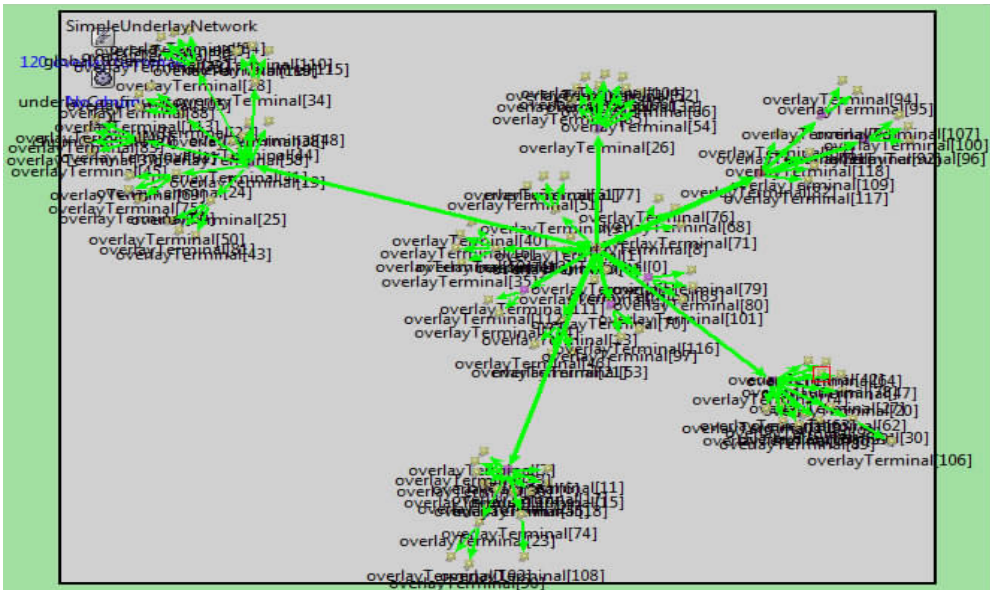


图 5 模拟仿真架构图

5.2 实验参数设置

下列是对实验中相关参数的设置情况：

(1) 节点的出度代表着节点的转发能力, 节点的最大出度表现为该节点能够容纳孩子节点的最大数目. 根据文献[30]和[31]中采用的实验参数, 本文在仿真实验中采用节点的最大出度 $deg_{\max}(v_i)$ 服从整数区间[2,4]之间的均匀分布.

(2) 应用层组播树中叶子节点的离开不会影响其它节点的数据传输. 而中间节点由于拥有若干子孙节点, 当其离开后将直接导致其所有子孙节点的数据传输中断^[32]. 因此, 本文在仿真实验中仅设置组播树的中间节点能够离开, 且中间节点的离开概率设置为 20%.

5.3 性能指标

节点的累计百分比/重加入时延指能够在某一时延内完成节点重加入操作的节点数目的占比. 假设时延为 t , 在 t 内能够完成重加入操作的节点数目为 m , 受影响节点的总数为 n , 则累计百分比/重加入时延 $CP(t)$ 的计算如式(3)所示.

$$CP(t) = (m/n) \times 100\% \quad (3)$$

节点的平均重加入时延指所有受影响节点重新加入到组播树所花费的时间的平均值. 假设节点离开时刻为 t_0 , 受影响节点 v_i 重加入到组播树中的时刻为 $t(v_i)$, 受影响节点的总数为 n , 则平均重加入时延 t_{ave} 的计算如式(4)所示.

$$t_{ave} = \frac{1}{n} \sum_{i=1}^n (t(v_i) - t_0) \quad (4)$$

5.4 仿真结果分析

将 HPLR 算法与 NICE 算法和 BFN 算法在累计百分比/重加入时延、平均重加入时延两方面进行对比.

图 6 统计了当节点数目为 500, 中间节点的离开概率为 20% 时, 节点的累计百分比/重加入时延. 在同一重加入时延内, 本文提出的 HPLR 算法的累计百分比值均大于 NICE 协议和 BFN 算法. 这意味着在某一时延内 HPLR 算法完成节点重加入操作的节点数目占比始终高于 NICE 协议和 BFN 算法. 当累计百分比达到 100% 时, HPLR 算法所需的重加入时延约为 1.15 s, 略小于 BFN 算法所需的 1.31 s, 并且明显小于 NICE 协议所需的 1.55 s.

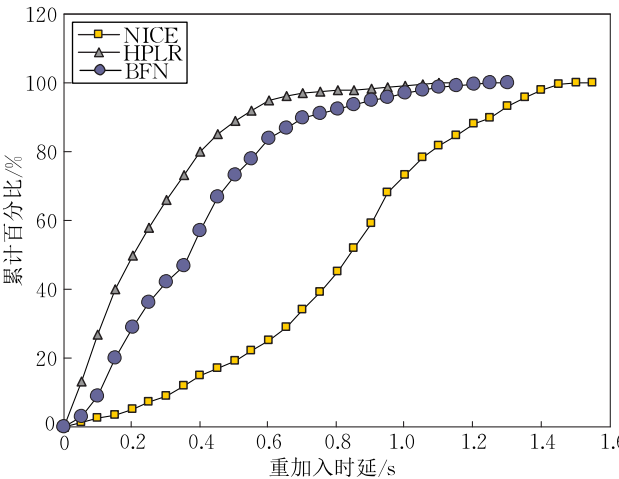


图 6 累计百分比/重加入时延

NICE 协议使用的数据传输树是基本路径树, 该树不是一棵基于约束的优化树, 当发生节点离开事件时, NICE 协议采取后向式的组播恢复策略, 即在节点离开组播树之后才开始制定应用层组播树的重构方案. 在 NICE 协议的恢复机制中, 当节点退出后, 受影响节点通过与仍在组播树中的成员节点进行联系, 再次加入组播树. 当节点的退出导致簇成员个数不满足条件时, 将会引发簇合并操作. 如果退出的节点为簇首, 则需要重新选取簇首节点. 后向式恢复、簇合并以及簇首的重新选择都在一定程度上增加了 NICE 协议的重加入时延.

BFN 算法为前向式恢复策略, 在节点离开之前先选择好备份父节点, 但当备份父节点已经没有剩余出度时, 需要在节点离开后进行重定向操作, 因此会对组播树的恢复速率造成一定的影响.

而 HPLR 算法采用基于分区的混合式组播恢复策略, 针对位于不同区域的节点的退出采用不同的恢复策略. 当位于中心区域的节点退出时, 运用前向式恢复策略, 即在节点离开组播树之前就做好恢复准备, 提前为受影响节点选取好备份父节点. 当位于边缘区域的节点退出时, 运用后向式恢复策略. HPLR 算法的混合式组播恢复策略能够有效地降低节点的重定向延时. 仿真数据显示, HPLR 算法在累计百分比/重加入时延方面较 NICE 协议和 BFN 算法具有明显的优势.

图 7 统计了平均重加入时延与组播规模的关系. 当组播规模在 100 个节点以内时, NICE 协议和 BFN 算法的平均重加入时延分别分布在 1.25 s 至 2.3 s 之间以及 1.05 s 至 1.4 s 之间, 而本文提出的 HPLR 算法的平均重加入时延则分布在 0.7 s 至

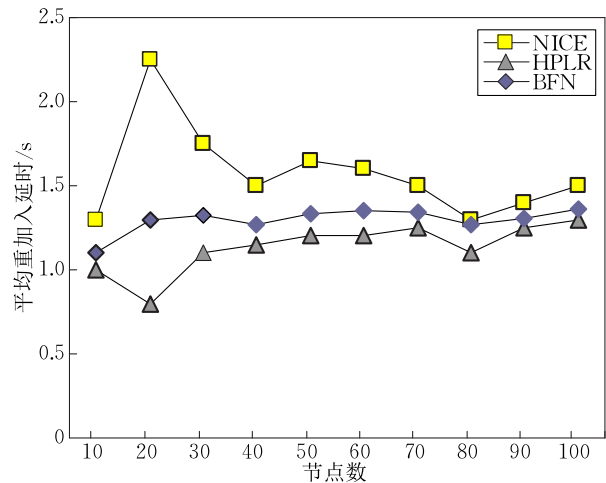


图 7 平均重加入时延

1.3 s 之间. 并且在同一组播规模下, HPLR 算法的平均重加入时延均小于 NICE 协议和 BFN 算法. 由于 HPLR 算法采用混合式的组播恢复策略, 对比仅采用后向式组播恢复策略的 NICE 协议以及仅采用前向式组播恢复策略的 BFN 算法而言, 能够有效地降低重加入时延, 且在 HPLR 算法中, 随着组播规模的增加其平均重加入时延在较小的范围内波动. 随着节点数目的增多, HPLR 算法和 NICE 协议的平均加入时延相差越来越小, 这是由于 NICE 协议适用于大规模用户, 当组播规模不断增大时, 这种优势更加凸显.

将 HPLR 算法与 Yang 算法在恢复后组播树的控制开销方面进行对比, 如图 8 所示. 控制开销指保存数据信息以及内存占用的开销. NICE 协议采取后向式的组播恢复策略; Yang 算法是一种前向式应用层组播树恢复算法. HPLR 算法采用基于分区的混合式组播恢复策略, 在加入时延方面, 与后向式的组播恢复策略更有可比性; 在控制开销方面, 与前向式恢复算法更有可比性.

图 8 统计了节点数目为 200 以内的恢复后组播树的控制开销. Yang 算法中, 当节点频繁加入或者离开组播树时, 非叶子节点都要重新执行 Yang 算法计算备用父节点, 增大了内存的占用以及保存数据信息所需的开销; HPLR 算法采用混合式的组播恢复策略, 组播树中位于中心区域和边缘区域的节点的相关信息均存放在集合中, 能够有效地降低维护组播树所需的控制数据包总量. 仿真数据显示, HPLR 算法在恢复后组播树的控制开销方面较 Yang 算法具有显著优势.

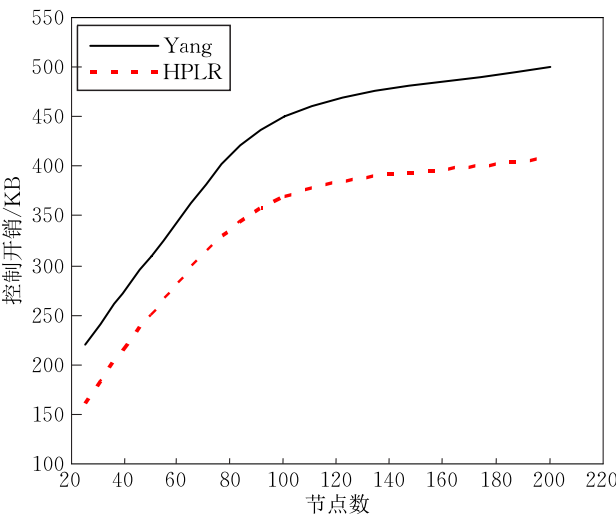


图 8 恢复后组播树的控制开销

6 结束语

本文针对当前应用层组播存在的稳定性问题,主要研究了在应用层组播树的中间节点失效之后如何高效地重构组播树的问题.本文的主要贡献为提出了一种混合的基于分区策略的应用层组播恢复算法 HPLR 算法.该算法首先提出了一种对组播树中的节点进行区域划分的方式,将节点的子孙节点的数目与其根路径长度的比值定义为该节点的服务能力,再根据节点服务能力的大小将应用层组播树划分为中心区域和边缘区域.在区域划分节点的基础上,本文分别对每个区域提出了相应的应用层组播树重构策略,以在系统的计算开销和时间开销方面达到平衡.针对中心区域节点失效的情况提出了一种前向式的组播树重构策略 HPLR-CD 算法;针对边缘区域的节点失效情况提出了一种后向式的组播树重构策略 HPLR-MD 算法.最后通过仿真实验,对比了本文 HPLR 算法与 NICE 算法和 BFN 算法在平均重加入时延、累计百分比/重加入时延这两方面的性能,以及本文算法与 Yang 算法在恢复后组播树的控制开销方面的性能,仿真结果表明本文提出的算法能够有效地降低组播树的恢复时延及控制开销.

本文对应用层组播的恢复策略进行了研究,但还存在诸多不足和有待继续完善的地方,今后将在以下几个方面展开研究工作.首先,将继续深入研究应用层组播的恢复策略,以获得更加高效且稳定的恢复算法.其次,对应用层组播领域存在的其他问题展开深层次的研究.除了本文介绍的应用层组播的稳定性问题之外,应用层组播还存在诸多其他问题.应用层组播由可信度不高的端主机转发数据,使其安全性问题更加引人关注,同时,应用层组播还在拥塞控制、流量控制、自私性等方面存在问题,下一步,我们将着重研究这些问题.

参 考 文 献

- [1] Deering S E. Multicast routing in Internetworks and extended LANs. *ACM Transactions on Computer System*, 1990, 8(2): 85-110
- [2] Chu Yang-Hua, Rao S G, Zhang-Hui. A case for end system multicast. *IEEE Journal on Selected Areas in Communications*, 2001, 20(8): 1456-1471
- [3] Chu Yang-Hua, Rao S G, Seshan S, Zhang-Hui. Enabling conferencing applications on the Internet using an overlay multicast architecture//*Proceedings of the ACM SIGCOMM*. San Diego, USA, 2001: 55-67
- [4] Diot C, Levine B N, Lyles B, et al. Deployment issues for the IP multicast service and architecture. *IEEE Network*, 2000, 14(1): 78-88
- [5] Cui Jian-Qun, Xiong Nai-Xue, Wu Li-Bing, Jia Ke-Ming. A novel source path topology map construction method in ALM. *Journal of Internet Technology*, 2015, 16(6): 1015-1022
- [6] Luo Jian-Guang, Zhao Li, Yang Shi-Qiang. An algorithm of constructing ALM tree based on user behavior analysis. *Journal of Computer Research and Development*, 2006, 43(9): 1557-1563(in Chinese)
(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. *计算机研究与发展*, 2006, 43(9): 1557-1563)
- [7] Cui Jian-Qun, Xiong Nai-Xue, Jong H P, et al. A novel and efficient source-path discovery and maintenance method for application layer multicast. *Computers & Electrical Engineering*, 2013, 39(1): 67-75
- [8] Pendarakis D, Shi S, Verma D, Waldvogel M. ALMI: An application level multicast infrastructure//*Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems*. San Francisco, USA, 2001: 49-60
- [9] Banerjee S, Bhattacharjee B, Kommareddy C. Scalable application layer multicast. *ACM SIGCOMM Computer Communication Review*, 2002, 32(4): 205-217
- [10] Tran D A, Hua K A, Do Tai. ZIGZAG: An efficient peer-to-peer scheme for media streaming//*Proceedings of the IEEE INFOCOM*. San Francisco, USA, 2003: 1283-1292
- [11] Padmanabhan V N, Wang J, Chou P A, Sripanidkulchai K. Distributing streaming media content using cooperative networking//*Proceedings of the 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video*. Florida, USA, 2002: 177-186
- [12] Chawathe Y D. Scattercast: An adaptable broadcast distribution framework. *Multimedia Systems*, 2003, 9(1): 104-118
- [13] Jannotti J, Gifford D K, Johnson K L, et al. Overcast: Reliable multicasting with on overlay network//*Proceedings of the Operating Systems Design & Implementation*. California, USA, 2000: 197-212
- [14] Zhang Bei-Chuan, Jamin S, Zhang Li-Xia. Host multicast: A framework for delivering multicast to end users//*Proceedings of the IEEE INFOCOM*. New York, USA, 2002: 1366-1375
- [15] Cao Ji-Jun, Su Jin-Shu. Delay-bounded and high stability spanning tree algorithm for application layer multicast. *Journal of Software*, 2010, 21(12): 3151-3164(in Chinese)
(曹继军, 苏金树. 应用层组播的时延受限高稳定性生成树算法. *软件学报*, 2010, 21(12): 3151-3164)
- [16] Banerjee S, Lee S, Bhattacharjee B, Srinivasan A. Resilient multicast using overlays. *IEEE/ACM Transactions on Networking*, 2006, 31(2): 237-248
- [17] Yang Meng-Kun, Fei Zong-Ming. A proactive approach to reconstructing overlay multicast trees//*Proceedings of the IEEE INFOCOM*. Toronto, Canada, 2004: 2743-2753

- [18] Kusumoto T, Kunichika Y, Katto J, Okubo S. Tree-based application layer multicast using proactive route maintenance and its implementation//Proceedings of the ACM SIGCOMM Workshop on Advances in Peer-to-Peer Multimedia Streaming. Philadelphia, USA, 2005; 49-58
- [19] Alpes P, Project-Team P, Eric P, et al. Application-level multicast transmission techniques over the Internet. IEEE Transactions on Device & Materials Reliability, 2004, 12(2): 462-469
- [20] Wong K-F S, Chan S-H G, Wong Wan-Ching, et al. Lateral error recovery for application-level multicast//Proceedings of the Joint Conference of the IEEE Computer & Communications Societies. Hong Kong, China, 2004; 2708-2718
- [21] Bawa M, Deshpande H, Garcia-Molina H. Transience of peers and streaming media. ACM SIGCOMM Computer Communication Review, 2003, 3(1): 107-112
- [22] Deshpande H, Bawa M, Garcia-Molina H. Streaming live media over a peer-to-peer network. Stanford Database Group Technical Report, 2001, 3(3): 75-77
- [23] Wang Jun-Yi, Wei Yan-Heng, Fu Jie-Lin, Zheng Lin. Highly stable scheme of application layer multicast in heterogeneous networks. Application Research of Computers, 2016, 33(1): 165-170(in Chinese)
(王俊义, 魏延恒, 符杰林, 郑霖. 异构网络环境下的高稳定性应用层组播方案. 计算机应用研究, 2016, 33(1): 165-170)
- [24] Lei Wen-Qing, Ma Cheng, Zhang Xin-Chang, Wang Lu. A SDN-like loss recovery solution in application layer multicast //Proceedings of the International Conference on Information Engineering for Mechanics and Materials. Shanghai, China, 2015; 852-857
- [25] Cui Jian-Qun, Zhang Wen-Lin, Huang Feng, Wu Li-Bing. Hierarchical adaptive recovery algorithm in mobile ALM//Proceedings of the Internet Conference of China. Shanghai, China, 2015; 95-105
- [26] Liao Xiao-Fei, Li De-Min, Chen Guang, et al. Low delay application layer multicast model based on tree-tree structure. Journal of Chinese Computer Systems, 2016, 37(11): 2493-2497(in Chinese)
(廖小飞, 李德敏, 陈光等. 一种基于树树结构的低时延应用层组播模型. 小型微型计算机系统, 2016, 37(11): 2493-2497)
- [27] Ye Yong-Jia. A Construction and Maintenance Scheme of Application Layer Multicast System Based on Degree-Constraints NICE Protocol in Mobile Environment [M. S. dissertation]. Central China Normal University, Wuhan, 2014(in Chinese)
(叶咏佳. 基于 NICE 具有度约束的移动应用层组播系统的构建及维护研究[硕士学位论文]. 华中师范大学, 武汉, 2014)
- [28] Zhang Xin-Chang, Yang Mei-Hong, Zhu Xiao-Jing, Luo Wan-Ming. A loss recovery approach for reliable application layer multicast. Journal of Systems & Software, 2012, 85(5): 1198-1204
- [29] Cui Jian-Qun, Lai Min-Cai, Jiang Wen-Bin, Chen Chuan-He. OverSim: A scalable application-layer multicast network simulation framework. Computer Engineering & Science, 2012, 34(10): 1-5(in Chinese)
(崔建群, 赖敏财, 蒋文斌, 陈传河. OverSim: 可扩展的应用层组播网络仿真框架. 计算机工程与科学, 2012, 34(10): 1-5)
- [30] Huo Lin, Li De-Shun, Tan Ying-Lu. Algorithms of spanning tree based on the stability probability and contribution link of nodes for application layer multicast. Journal of Computer Research & Development, 2012, 49(12): 2559-2567 (in Chinese)
(霍林, 李德顺, 谭颖璐. 基于节点稳定概率和链路贡献度的应用层组播树生成算法. 计算机研究与发展, 2012, 49(12): 2559-2567)
- [31] Fei Z, Yang M. Restoring delivery tree from node failures in overlay multicast. IEITC Transactions on Communications, 2005, 88(5): 2046-2053
- [32] Zhang Wen-Lin. The Research of Minimum Delay Tree Constructing Algorithm and Hierarchical Recovery Strategy in Mobile ALM [M. S. dissertation]. Central China Normal University, Wuhan, 2015(in Chinese)
(张文琳. 移动应用层组播最小时延树生成算法及分层恢复策略研究[硕士学位论文]. 华中师范大学, 武汉, 2015)



CUI Jian-Qun, born in 1974, Ph.D., professor, Ph.D. supervisor. Her current research interests include quality of network service, network management, high-performance computing and application layer multicast.

CHENG Ai-Ling, born in 1990, M.S. Her current research interests include quality of network service and application layer multicast.

HAN Jie, born in 1986, M.S. Her current research interests include quality of network service and application layer multicast.

CHANG Ya-Nan, born in 1984, Ph.D., lecturer. Her current research interests include wireless mesh networks, network management and application layer multicast.

WU Li-Bing, born in 1972, Ph.D., professor, Ph.D. supervisor. His current research interests include network communication, grid computing and Internet of Things.

Background

Traditional unicast communication mode needs to create a separate data transmission channel between the client and the server, but the model cannot meet the requirements of the current multimedia data on the transmission efficiency. The emergence of IP multicast solves the problem of low efficiency, this is because the IP multicast changed the Internet based on the design principle of unicast, which needs to change the existing network infrastructure, leading to the slow progress in global deployment.

Due to the defects existed in the IP Multicast which limits the development of its own, Francis and Zhang separately proposed the Application Layer Multicast (Application Layer Multicast, ALM) in 1997 and 1998. Its characteristic is that without changing the existing network infrastructure, the terminal system can implement the functions of data forwarding and replication. So the ALM is easy to deploy. So far, there have been a lot of application layer multicast protocols successfully applied, for example, YOID, Narada, HMTP, NICE, etc.

This paper mainly studies Application Layer Multicast aiming at the high stability and low latency of application layer multicast tree construction algorithm and high efficiency of application layer multicast tree recovery algorithm. This paper presents a hybrid partition based method for loss

recovery (HPLR) in ALM. In this method, we redefine the server ability of node, and divide the tree into two parts according to node's server ability. Each part of the tree has its own recovery method respectively, so as to achieve a balance between computational overhead and time overhead. Aiming at the failure of the central area node, this paper proposes a forward-based multicast tree reconstruction strategy HPLR-CD. In this algorithm, the backup parent node is searched for its children before the node fails, and the affected node contacts the backup parent node to recover the data transfer. Aiming at the failure of the margin area node, this paper proposes a backward multicast tree reconstruction strategy HPLR-MD algorithm. When the node fails, the affected node is in contact with the grandfather node to restore the multicast connection, thus restricting the change of the topology caused by the node exit event to the multicast tree in the local scope.

The simulations demonstrate that the HPLR algorithm can effectively reduce the rejoin time and control overhead after the nodes leave. This paper was supported by the National Natural Science Foundation of China (61370108, 61672257, 61170017, 61272112) and the Natural Science Foundation of Hubei Province (2016CFB145).