度约束应用层组播系统的设计与实现①

Design and Implementation of Application Layer Multicast System with Degree-Constrain

周学芝　黄东军　(中南大学 信息科学与工程学院 湖南 长沙 410083)

摘　要： 提出了一个具有度约束特性的应用层组播树构造算法，采用节点可用网络带宽、CPU 占用率、可用存储空间、当前进程数等描述节点度数，通过在源根组播树构造算法中引入节点的度约束，旨在生成负载更均衡、整体性能更优的组播覆盖网。采用 Java 语言实现了一个基于该算法的群组通信系统原型，在满足主机容量限制的前提下，成员节点可以任意地加入或退出群组，并实时传输字符数据流。实验表明，具有度约束的组播算法，能有效降低成员节点计算能力波动对组播系统的冲击，减少平均延时，提高传输吞吐量。

关键词： 应用层组播 度约束 延时 实现 评价

1　引言

近年来，为了缓解网络带宽瓶颈，组播(Multicast)技术越来越受到重视。由于 IP 组播需要改变现存路由器的无状态特性，导致管理控制困难，可扩展性差，因而至今不能在互联网中得到广泛应用。将复杂的组播功能放在端系统实现被证明是有效的，应用层组播基于“端到端的思想(End to End Argument)”[1]，在分布式系统的设计中，把复杂的应用功能放在网络的边缘，即组播服务在终端主机而不是在网络路由器中实现，让网络核心部分只做最通用的数据传输而不实现特殊应用。这样就有效地降低了核心网络的复杂性，便于升级维护，同时提高网络的通用性和灵活性，在增加新应用时不必改变核心网络。

但是，要设计出高效的应用层组播系统也面临新的挑战，研究人员需要考虑许多不同于网络层的因素。例如，怎样使由端系统连接而成的组播覆盖网尽可能地接近网络层的实际传输结构，怎样降低主机频繁加入或退出对组播稳定性的冲击等，此外，应用层组播仍然要解决可扩展、可管理等问题。这些在以往的研究中都得到了重视[2-4]。然而，一个被忽视但同样重要的因素是主机的容量限制，以往的研究基本没有考虑主机容量的有限性[5,6]，认为主机具有足够的容量(即能与任意多的主机进行连接)。但事实并非如此，主机能直接连接的邻居数目总是有限的。如果不考虑主机的容量限制，就有可能使某个节点连接过多的邻居，从而在该主机处产生瓶颈，导致整个组播系统性能的急剧下降。为此，本文提出了一个具有度约束特性的应用层组播树构造算法，它采用节点的可用网络带宽、CPU 占用率、可用存储空间、当前进程数等参数描述节点容量 (度数)，通过在源根组播树算法中引入节点的度约束功能，试图消除节点连接过多邻居的倾向，生成负载更均衡、整体性能更优的组播覆盖网。本文还使用 Java 语言，在 Eclipse3.3+JDK6.0 平台上，实现了一个基于该算法的群组通信系统原型。实验表明，与无度约束的同类组播算法比较，本系统能有效降低成员节点计算能力波动对整个组播系统的冲击，减少平均延时，提高传输吞吐量。

2　算法设计

2.1 基本思想

为了研究度约束对组播性能的影响，需要首先确定一种基本的组播树构造算法。在应用层，组播算法

① 基金项目:国家自然科学基金项目(60873188)
收稿时间:2008-09-01

可分为基于 P2P 覆盖网的和基于树结构的两类。前者要求生成一个 P2P 覆盖网，然后基于该覆盖网实施组播。后者则直接由端系统建立树状传输结构(可称为基于覆盖树的组播算法)。本文采用效率较高的后者，并且限于源根树算法(Source-specific Tree)。之所以这样选择，是因为源根树算法最能体现节点的度数对组播系统的影响，而在基于 P2P 覆盖网的组播算法中，度约束应当在覆盖网建立阶段而不是组播阶段考虑。在基于覆盖树的组播算法中，与源根树算法对应的共享树技术因汇聚点的存在而天然具有瓶颈效应，度约束无法实施。

图 1 表示组播树覆盖网建立的会话过程。其中 New_Node 表示要加入群组的新成员节点，DS 是目录服务器，实心的节点和粗线条表示当前组播树。

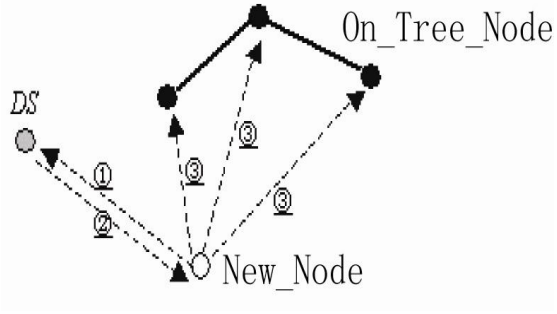

图 1　组播覆盖网建立会话过程

算法的基本过程如下：首先由欲加入组播树的接收节点 New_Node 访问目录服务器 DS（图中用①表示），DS 返回群组成员地址信息（图中用②表示），接着 New_Node 运用测量技术探测到所有当前在树节点的连接性能（图中用③表示）；最后从这些连接中选择延时最小、带宽最大的一个加入组播树。

上述算法简单有效，并且考虑了连接的延时和带宽等 QoS 需求，但是没有度约束功能，即一个节点可以连接的邻居数不受限制，这会导致一些节点因连接过多邻居而出现瓶颈效应，使组播树的整体性能急剧下降。可见，有必要在算法中增加度约束功能。

为此，我们在前述算法的第③步中，拟让 New_Node 不仅探测自己到各个在树节点的延时与带宽，还请求这些在树节点计算各自的度数，并返回计算结果。这样，New_Node 就可以比较各个候选父节点的度数，把度数和延时、带宽等综合起来考虑，从中选择综合性能最佳的节点连上去。

2.2 度数的计算

度数计算是实现度约束的关键。传统方法过于简单，仅仅考虑了节点的可用带宽[7][8]，即可用带宽除以请求带宽，所得商即为可连接的邻居数，例如，如果一个在树节点的可用带宽是 10 个单位，而请求带宽是 2 个单位，则该节点可连接的邻居数是 5。显然,这个启发式存在问题，因为它会导致这样一种现象,即某个 CPU 占用率较高、可用内存较少的节点由于处在高速网络内而有较大的可用带宽从而连接了较多的邻居，但实际上其计算能力已经无法再处理真正的组播数据流了。可见，在考虑一个节点的连接能力时，不仅要把可用带宽计算在内，还要考虑 CPU 占用率、可用内存等因素。

本文设计的度数计算方法如下：首先，把一个节点当前的可用网络带宽(N Kbps)、CPU 占用率(C)、可用内存大小(M MB)、系统进程表中的进程数(P)作为计算度数的基本参数。其次，由于接收节点仅关心其请求带宽是否得到满足，因此我们把可用带宽除以请求带宽的商作为一个节点的基本度数。最后，我们设计新节点选择其父节点的启发式：该启发式规定，所有候选父节点首先按下式的综合评价值排队：

$$Degree = (1 - C) * \frac{M}{P}$$

上式表明，CPU 占用率低、可用内存大、系统进程数少的节点将排在前列；然后，从前往后选择基本度数最大者，作为正式父节点。

上述启发式贯彻了这样的原则：可用带宽是最关键的因素，同时要兼顾节点的计算能力。在相同基本度数的条件下，那些综合性能更好的节点将被选择。这样，我们就得到了新的具有度约束能力的组播树构造算法。

2.3 节点的退出

组播通信的一个特点就是节点退出群组后组播树要立即重构。节点退出有两种情况：正常退出和异常退出，必须针对这两种情况分别设计重构算法。

(1)节点正常退出

节点退出过程如图 2 所示，以节点 J 退出为例。首先，节点 J 向 DS 发送退出报文，DS 接收请求后，获取 J 的父节点 I，以及 J 的所有子节点(K 和 H)。设置 J 的子节 K 为 I 的直接子节点。同时，将 K 下所有的子节点的层数减 1。而 H 将作为其它度数未满的节

点的子节点，并更新 H 的所有子节点的层数。

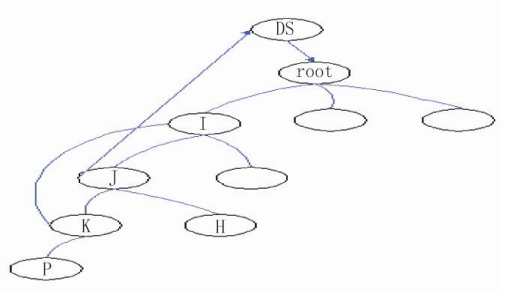

图 2　节点的正常退出

在处理退出请求时，必须首先重构覆盖网，再删除退出节点的信息，否则会引发错误。

(2)节点异常退出

在任何网络应用中，故障都是不可避免的，因此，系统必须给这些突如奇来的故障提供一些处理的方法，避免系统由于这些因素的存在而崩溃。在本系统实现中，我们定义了很多错误代码，每产生一个错误就会对应一个错误代码。分析这些代码就能得出错误的原因,并调用不同错误处理方法来处理这些故障。节点的异常退出就是其中一种。在这里，我们应用心跳算法来轮询这种故障,一旦检测到有节点异常退出，系统就会启动处理正常退出的重构算法。心跳算法如下：

①在 DS 中创建心跳算法的后台线程；

②每隔 120 秒，DS 向所有终端发送一带特定标识的数据包；

③终端在收到这一数据包后，必须做出回应，即向目录服务器 DS 发送应答数据包；

④DS 接收终端的应答数据包，并记录终端的 IP，保存在数组 IPs 中；

⑤如果某 IP 不在当前 IP 列表中，调用覆盖网重构算法，对该节点的退出进行组播网重构；

⑥向所有在线的终端泛洪新的组播网信息；

3　系统的设计与实现

3.1 系统结构

这是一个应用层组播通信系统原型，具有构建覆盖组播树、组播字符流、树重构等功能。系统结构如图 3 所示，它采用分层结构，最上层提供统一封装的用户操作界面；中间层为响应用户操作的接口，在该层中，用户操作将转化为系统响应；第三层为系统的功能层，提供应用层组播系统的所有功能，是系统的核心部分；最下层为网络协议层，本系统是基于 TCP/IP 协议的应用层组播系统。

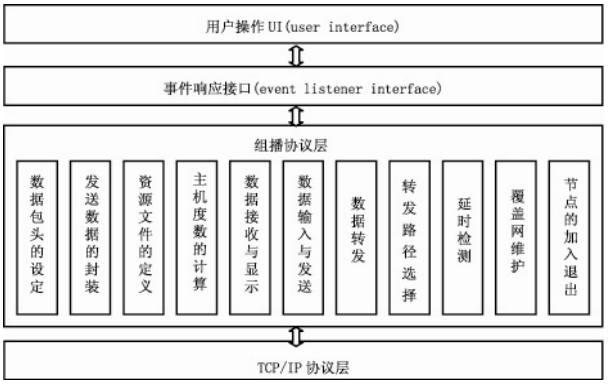

图 3　系统层次结构图

用户操作 UI 为用户提供统一的可视化操作界面，用户通过鼠标、键盘的操作，使系统调用相关功能。

事件响应接口层在系统中扮演控制器的角色，每当用户有鼠标点击或键盘按下的操作，系统将其转化为各种消息，事件响应接口在接收到这些消息后，对不同的消息做出不同的响应。

第三层称为组播协议层，是本系统中的核心部分，所有重要的功能都在这一层实现。

数据包头设定：在本系统，涉及到多种信息的发送和接收，因此，必须要有一个数据头来标识发送信息的类型，接收者根据接收到的消息头，对数据包进行相应处理。

发送数据的封装：在数据接收和发送前，必须将消息体和各种消息控制信息封装在一个可被操作的数据结构中。

资源文件的定义：软件可扩展的思想就体现在这一点上，各种常量数据都在资源文件中，如界面上显示的字符，通信端口的设定等，如果需要改变界面风格，只需要更改资源文件的内容即可，不需更改程序。

主机度数的计算：按照 2.2 节中给出的度数定义计算。

数据的接收与显示：终端可以接收到其它源端发送的组播信息，并在界面上显示，显示的具体内容包括发送方的主机名，IP 地址，发送时间和发送的内容。

数据的输入与发送：源端可以在发送输入框中输入要发送的内容，然后组播到所有接收者。

数据转发：应用层组播是一个依靠端系统复制和转发数据的系统，各个节点根据组播转发表决定的转

发路径发送数据。

转发路径选择：根据组播转发表决定转发路径。

延时检测：利用探测数据包和响应数据包，检测本节点到候选父节点的延时。

覆盖网维护：在节点加入或退出组播时，覆盖网信息需要更新，这时需要通知所有节点新的网络信息。

节点的加入与退出：节点加入和退出时需要对组播网进行重构。

3.2 系统功能的实现

我们使用 Java 语言，在 Eclipse3.3+JDK6.0 平台上实现基于度约束算法的群组通信系统原型。其中，配置资源文件用到 Native2Ascii 工具，将汉字转化为 unicode 码。通过编写本地方法，获取系统硬件资源信息，如 CPU 占用率、可用内存大小、可用网络带宽和当前进程数，编译成 dll 文件供系统调用。

系统启动时，首先需要初始化，获取当前节点的信息，通过调用 dll 文件中的本地方法计算主机的度数，获取主机 IP、主机名、登录时间。接着创建接收线程，计算机在 9000 号默认端口监听接收信息。然后初始化组播覆盖网的节点数组，将其当前节点信息添加到数组中。

各个节点的界面如图 4 所示，左上区域用来显示群组成员的基本信息，如主机名、IP 地址等；左下区域供用户输入要发送的字符数据；右上区域显示统计数据，如群组主机数等；右下区域有待扩展，如增加加密功能等。

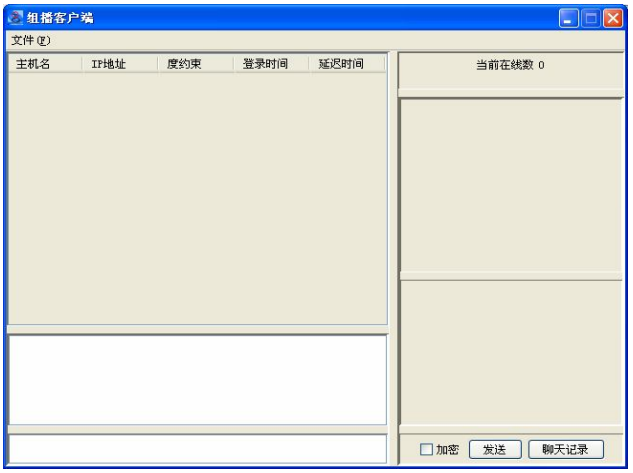

图 4 系统初始化界面

这里还需要特别说明 DS 的实现方法。通常，DS 是一台独立服务器，但是本文采用一种简便方法：由源端充当 DS。这种方法更符合源根组播的实际，因为源根组播是由源端发起的，所以它是第一个在树节点，这样其他节点均可以在源端注册，从而实现群组信息的会聚。

3.3 系统评价

我们在局域网环境下对系统进行了反复测试。从功能上看，系统达到了设计目标，能够实现源组播通信。从性能上看，我们在 70 次测试中，通过逐步增大通信量（吞吐量），考察从源端到各个终端的平均传输延时，发现该值的增加非常平缓。这说明系统均衡负载的能力较强，出现瓶颈的概率 较低。作为对比，我们在系统中去掉度约束功能，而其他功能保持不变，在同样的 70 次测试中，发现平均延时比前者增加约 12%。这说明度约束功能发挥了明显的作用。

4 结束语

本文旨在考察度约束功能的效用。我们首先从理论上提出一个具有度约束功能的应用层组播算法，然后通过编程实现系统原型，从实践上验证该算法的作用，达到了预期目标。下一步我们计划扩充系统功能，实现音视频数据流的传输，同时还将研究系统的安全性、稳定性等问题。

参考文献

1 李伟,沈长宁.应用层组播协议研究.计算机工程与应用,2004,24(4):156－159.

2 罗建光,赵黎,杨士强.基于用户行为的应用层组播树生成算法.计算机研究与发展,2006,43(9):1557－1563.

3 许建真,濮松兰,张福炎.高效的基于平面的层次化应用层组播树模型.计算机科学,2008,35(6):67－70.

4 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.

5 马炫,孙丽敏,张亚龙.度约束 QoS 组播路由遗传算法.计算机工程与应用,2007,43(9):114－116.

6 王德志,余镇危,甘金颖,等.基于免疫克隆的带度约束的应用层组播路由算法.计算机工程,2007,33(3):105－107.

7 潘耘,余镇危,王行刚,等.Overlay 组播路由的负载平衡问题.电子与信息学报,2007,29(3):739－742.

8 Zhang BC, Jamin S, Zhang LX. Host Multicast: A Framework for Delivering Multicast To End Users. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002). New York, 2002,3:1366－1375.