第35卷 第18期 计 算 机 工 程 2009年9月
Vol.35 No.18 Computer Engineering September 2009
·网络与通信· 文章编号：1000—3428(2009)18—0105—03 文献标识码：A 中图分类号：TP393

双端协作的P2P流媒体节点选择机制

孙名松[1]，张潇依[2]，姚 亮[2]
(1. 哈尔滨理工大学网络信息中心，哈尔滨 150080；2. 哈尔滨理工大学计算机科学与技术学院，哈尔滨 150080)

摘 要：针对P2P流媒体系统现有节点选择方法仅在请求节点执行且功能单一的问题，提出超级节点和请求节点双端协作的节点选择机制。在超级节点端，根据路由相似的思想提出面向网络拓扑聚集性的候选节点选择算法。在请求节点端，综合考虑系统动态性与异构性，提出基于枚举的服务节点选择算法，保证流服务质量，融入动态冗余思想以提高健壮性。仿真结果表明，该机制能更好地改进系统的网络拓扑聚集性，提高流服务质量和健壮性。

关键词：P2P流媒体；节点选择；双端协作；路由相似；枚举；动态冗余

Double-ended Cooperation Peer Selection Strategy for P2P Media Streaming

SUN Ming-song[1], ZHANG Xiao-yi[2], YAO Liang[2]
(1. Network Information Center, Harbin University of Science & Technology, Harbin 150080;
2. College of Computer Science & Technology, Harbin University of Science & Technology, Harbin 150080)

【Abstract】 In P2P media streaming system, the existing peer selection methods only focus on the implementation of the requesting peer, and the function is unitary. In order to resolve these problems, a double-ended cooperation peer selection strategy is proposed, which is between super peer and requesting peer. At the super peer, an algorithm of selecting candidate peers based on routing resemblance is proposed, which is topology conglomeration oriented. At the requesting peer, considering the dynamic and heterogeneity of the system, an algorithm of selecting service peers based on enumeration is proposed, which can guarantee the streaming QoS and meanwhile improve the robust by adding the thought of dynamic redundancy. Simulation results show that the strategy can improve the topology conglomeration, streaming QoS and robust much better.

【Key words】 P2P media streaming; peer selection; double-ended cooperation; routing resemblance; enumeration; dynamic redundancy

1 概述

节点选择是P2P流媒体系统中的一项关键技术，现有节点选择方法大多仅在请求节点端执行，忽视了超级节点端候选节点的选择质量，且只改进了系统的单方面性能，功能比较单一。在现有系统中，AnySee[1]使用路标标注的方法解决网络拓扑聚集性问题；BigMedia采用运营商与地区相结合的二维向量与IP映射的方法；文献[2-3]讨论了基于坐标的方法。前2种方法本质上都是要维护一个节点与其位置信息的映射表，这张表的覆盖面、准确性与有效性难以保证；而基于坐标的方法复杂性又过高。本文提出超级节点和请求节点双端协作的节点选择机制，根据路由相似思想[4]提出一种无须维护任何映射关系、复杂性低的候选节点选择算法。

2 混合式P2P网络拓扑结构

本文提出一种P2P和传统C/S相结合的混合式P2P网络拓扑结构，如图1所示，主要由流媒体服务器、超级节点层和普通节点层3部分组成。

(1)流媒体服务器(Media Server, MS)：媒体文件的分发源。

(2)超级节点(Super Peer, SP)：比普通节点拥有更高性能的处理器，更大的网络带宽和负载能力。主要有2个功能：1)维护其孩子节点(普通节点)的信息列表，存储其孩子节点的共享资源索引；2)接受其孩子节点的资源请求。

(3)普通节点(Ordinary Peer, OP)：即请求节点。

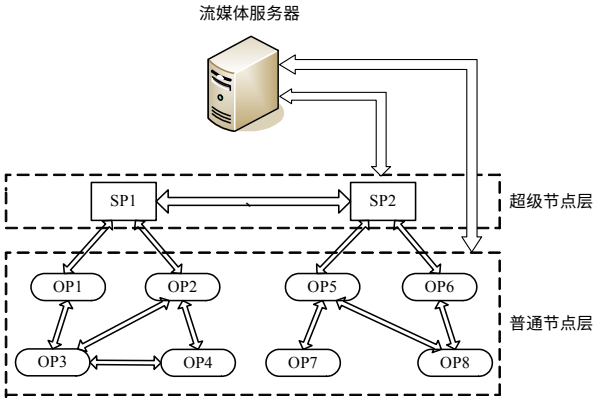


图1 混合式P2P网络拓扑结构

3 双端协作节点选择机制

双端协作节点选择机制执行流程如图2所示。请求节点发出资源请求后，资源搜索子系统负责查找资源节点列表，在超级节点端对资源节点列表进行快速初步筛选，挑选出候选节点列表返回给请求节点。在请求节点端对候选节点列表进行进一步筛选，挑选出优秀的服务节点集提交给流媒体应用系统。

作者简介：孙名松(1963－)，男，教授、博士，主研方向：网络安全，网络应用；张潇依、姚 亮，硕士研究生

收稿日期：2008-12-16 **E-mail：**zhang_xiaoyi@163.com

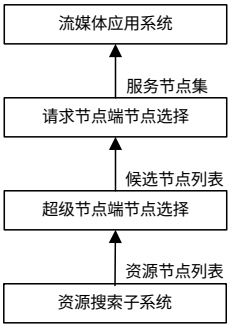


图 2　双端协作节点选择机制执行流程

3.1　超级节点端节点的选择

3.1.1　节点选择目标

考虑到超级节点返回给请求节点的候选节点列表大小，太大会增加请求节点进一步选择优秀服务节点集的复杂度；太小又可能会导致候选节点不够用。文献[5]通过试验分析证明：当超级节点返回的候选节点列表大小为 20 时，即可满足请求节点进一步选择优秀服务节点集的需要。据此，确定超级节点端节点选择目标：对资源搜索子系统查询到的庞大的资源节点列表进行快速初步筛选，挑选出与请求节点位于同一区域同一 ISP 内的 20 个候选节点。

3.1.2　算法描述

路由相似思想：位于同一区域同一 ISP 内的 2 个节点 A 与 B，分别到达与其相隔较远的另一目标节点 C，那么 $A\rightarrow C$ 和 $B\rightarrow C$ 这 2 条路径上经过的中间路由器绝大部分都是相同的。

基于该思想提出算法，具体描述如下：

(1)请求节点 R 向超级节点发送资源请求，同时汇报自己所在区域及 ISP。

(2)超级节点根据 R 所在区域及 ISP，选择一个与 R 相隔较远的节点 D 作为目标节点(如中国北京的网通用户，可选择美国纽约的节点作为目标节点)，并使用路由跟踪工具得到 $R\rightarrow D$ 的路由(其中包含途经所有路由器的 IP)。

(3)超级节点从资源搜索子系统查询到的资源节点列表中，随机取节点 X，使用路由跟踪工具得到 $X\rightarrow D$ 的路由(其中包含途经所有路由器的 IP)。

(4)超级节点将 $X\rightarrow D$ 的路由与 $R\rightarrow D$ 的路由进行比较，若有大于 50%的路由器 IP 不同，则认为 X 与 R 位于不同区域不同 ISP 内，将 X 过滤掉；否则认为 X 与 R 位于同一区域同一 ISP 内，将 X 加入到候选节点列表中。

(5)重复(3)与(4)中的操作，直到候选节点列表中有 20 个候选节点时停止。

(6)将选出的候选节点列表返回给请求节点。

3.1.3　算法复杂度分析

空间复杂度上，该算法只须增加 2 条路由的存储空间，因此，在空间复杂度上，它是可行的。

在时间复杂度上，主要从以下 3 个方面进行说明：

(1)该算法仅在请求节点请求资源的时候执行。

(2)设资源节点列表的大小为 n，每条跟踪路由途径的路由器数量相对来说较小，可认为是常量阶，那么该算法的时间复杂度为 $O(n)$。另外，由于该算法只要选出 20 个候选节点就停止，因此无须处理全部 n 个资源节点，并且对于每个资源节点，只要发现有 50%的路由器 IP 不同就停止比较，无须比较整条路由。在实际运行时，该算法所需的运行时间将非常小。

(3)超级节点拥有高性能的处理器，其处理能力比普通节点强很多，这将进一步缩短该算法的运行时间。

根据上述分析，可以认为该算法在时间复杂度上也是可行的。

3.2　请求节点端节点的选择

3.2.1　节点选择目标

考虑到在实际系统中，单个节点提供数据流往往不能满足播放要求，因此，选择多个服务节点同时并行提供数据。

为保证流服务质量，综合考虑候选节点以及候选节点与请求节点间链路的动态变化与性能差异。设流媒体播放速率为 V_0，请求节点 R 从候选节点 P_i 获得的下载速率为 V_i，请求节点 R 与候选节点 P_i 间链路的丢包率为 L_i，考虑到候选节点进出系统的动态性及系统的负载均衡性，为每个候选节点设置 2 个参数：可用性 A_i 和可连接性 C_i，分别定义为

$$A(i)=\frac{\sum_{i=1}^{n}\frac{t_i}{24}}{n} \tag{1}$$

其中，t_i 表示节点 P_i 在一天内可提供的下载时间；n 表示取最近 n 天进行观测。

$$C(i)=\frac{C_{\max}-C_{\text{connected}}}{C_{\max}} \tag{2}$$

其中，$C_{\max}$ 表示每个节点可连接的并发流的最大数量；$C_{\text{connected}}$ 表示节点 P_i 当前已经连接的并发流数量。

将节点 P_i 的优良度 $g(i)$定义为

$$g(i)=A(i)\times C(i)\times(1-L_i) \tag{3}$$

设最终选出的服务节点集中共含有 m 个节点，则在接收端接收到的聚合流速率 $V_{\text{aggregated}}$ 可表示为

$$V_{\text{aggrerated}}=\sum_{i=1}^{m}V_i\times g(i) \tag{4}$$

根据上述设定，确定请求节点端节点选择目标：从超级节点返回的候选节点列表中(含 20 个候选节点)，进一步挑选出优秀的服务节点集，使聚合流速率 $V_{\text{aggregated}}$ 最大(假定接收端有无限的接收能力)。

3.2.2　约束条件

综合考虑流媒体系统的特点，须设定如下约束条件：

(1)服务节点提供的下载速率之和应不小于流媒体播放速率，才能保证媒体流的流畅度高。

$$\sum_{i=1}^{m}V_i\geq V_0 \tag{5}$$

(2)考虑到服务节点集如果太大，请求节点需要同时维持的并发流数量就会太多，这将增加其很多的额外开销，因此，服务节点集的大小 m 应据经验设置一个最大值 $M_{\max}$。

$$m\leq M_{\max} \tag{6}$$

3.2.3　数学模型

综上分析，建立数学模型如下

$$\text{Max } V_{\text{aggrerated}}=\sum_{i=1}^{m}V_i\times g(i) \tag{7}$$

$$\text{st.}\begin{cases}\sum_{i=1}^{m}V_i\geq V_0 & (8)\\ m\leq M_{\max} & (9)\end{cases}$$

3.2.4　算法描述

本文采用枚举思想，具体算法描述如下：

(1)从候选节点列表中枚举出所有满足式(9)的子集。

(2)对 Step1 中得到的所有子集，逐个用式(8)进行检验，

将不满足式(8)的子集过滤掉，满足式(8)的子集保留，即为同时满足式(8)和式(9)的所有子集。

(3)对(2)中得到的子集，逐个按式(4)计算其聚合流速率，并将计算结果从大到小排序，则其中聚合流速率最大的一个子集即为要选择的服务节点集。

(4)将选出的服务节点集返回给流媒体应用系统。

3.2.5 节点失效处理

为了在发生节点失效时，保证流会话能够继续进行而不中断，提高系统的健壮性，提出动态冗余思想。动态冗余指通过重复配置系统的一些部件，当系统发生故障时，重复配置的部件立即介入，承担故障部件的工作，与此同时，从系统中删除掉故障部件并且据系统此刻的状态动态产生新的重复部件，使得系统中每时每刻都同时有主从 2 套部件可用，由此提高系统健壮性的一种技术。与一般意义上的冗余相比，其特点与优势主要体现在动态上，即系统中的主从 2 套部件不是固定不变的，而是随系统变化而动态变化的。

只须对前文算法做出如下改进，即可将动态冗余思想融入其中，使得系统中每时每刻都同时有主从 2 个随系统变化而动态变化的可用节点集：

(1)第 1 次执行算法时，在选出服务节点集的同时，将聚合流速率次大的子集作为备用节点集。

(2)每次发生节点失效时，都进行如下操作：立即启用备用节点集代替服务节点集提供服务，并将失效的服务节点集从系统中删除，同时，按照失效时的各种参数值重新计算式(4)，将计算结果重新排序，选择其中最大的一个子集作为新的备用节点集。

3.2.6 算法可行性分析

算法可行性分析如下：

(1)经超级节点的初步筛选后，该算法的输入候选节点数量很少，只有 20 个，这使问题的复杂性大大降低。

(2)该算法只在流会话开始时和发生节点失效时才执行，并且由于服务节点是经过精心设计筛选的，能在很大程度上保证其有效性，发生失效的概率非常小，因此该算法执行的次数非常少。

(3)目前即使是普通 PC 机，也已具备了非常强的处理能力，在非常短的时间内完成该算法是完全可能的。

4 仿真实验与分析

为了更好地说明本文提出的节点选择机制的有效性，分别测试双端协作节点选择算法(DE-Coordination)以及目前 2 种代表性的节点选择算法带宽优先算法(Bandwidth First)和随机选择算法(Random)对系统性能的影响。

4.1 实验环境

通过修改 P2P 仿真软件 PeerSim 对 P2P 流媒体系统进行仿真，通过控制节点以柏松分布进入系统，并使节点在系统中的服务时间为指数分布来模拟 P2P 系统节点的动态特性。覆盖网节点总数在 0~1 000 之间变化，同时将覆盖网划分为 16 个 ISP 服务区域。流会话将持续 60 min，流速率 R_0 为 1 Mb/s。

4.2 评价指标

(1)网络拓扑聚集性：通过覆盖网压力来衡量。覆盖网压力定义为通过覆盖网进行数据分发时，穿越骨干网的流量与所有节点下载流量总和的比值。

(2)流服务质量：采用接收端实际获得的聚合流速率来衡量。

(3)健壮性：采用在发生节点失效的情况下，接收端获得的聚合流速率的变化情况进行衡量。

4.3 实验结果与分析

4.3.1 网络拓扑聚集性分析

覆盖网节点总数在 0~1 000 之间变化时，基于 3 种节点选择算法的覆盖网压力变化的曲线如图 3 所示。由此可知，带宽优先算法和随机选择算法的覆盖网压力过大，这是因为它们都没有考虑系统对 ISP 的影响，没有考虑优先选择同一地区同一 ISP 内的节点，所以造成了大量穿越骨干网的流量。双端协作节点选择算法随着节点数的增加，其覆盖网压力快速下降。

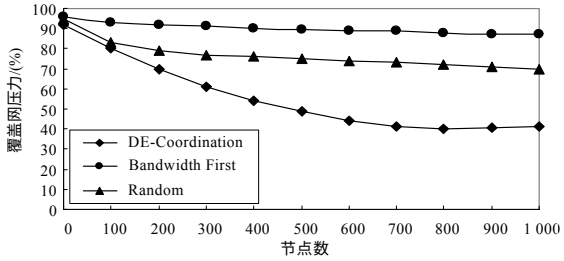


图 3 节点数与覆盖网压力关系

4.3.2 流服务质量分析

图 4 显示了在整个流会话期间，3 种节点选择算法的接收端实际获得的聚合流速率。可以看出，双端协作节点选择算法的聚合流速率最高，它能够提供 1.035 Mb/s 左右比较稳定的聚合流速率，对于 1 Mb/s 的播放速率，完全能够保证高质量的播放。带宽优先算法的聚合流速率比双端协作算法的稍低，而随机选择算法的聚合流速率最低，这是因为随机选择算法没有考虑到候选节点以及候选节点与请求节点间链路的动态变化与性能差异，所以无法保证节点的流服务质量。

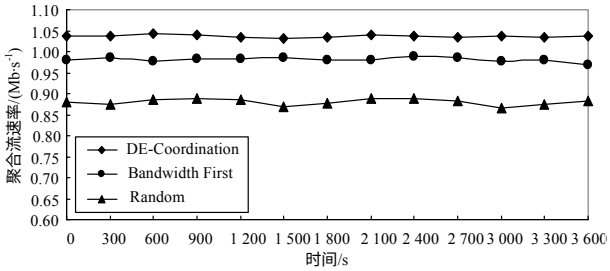


图 4 接收端实际获得的聚合流速率

4.3.3 健壮性分析

为了测试系统的健壮性，对节点失效进行仿真，在整个流会话期间，随机设置 3 次节点失效。图 5 显示了在有节点失效的情况下，3 种节点选择算法的接收端获得的聚合流速率的变化情况。

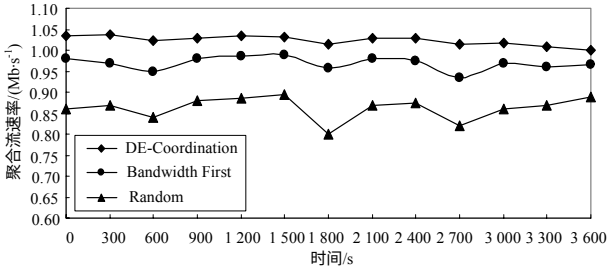


图 5 节点失效时接收端获得的聚合流速率

(下转第 110 页)