# P2P 视频直播系统中的分布式负载均衡算法①

## A Distributed Load Balancing Arithmetic in Video Living Broadcast System Based on P2P

崔虹燕 (兰州商学院 信息工程学院 甘肃 兰州 730020)

**摘 要:** 在分析了 P2P 网络应用模式优势的基础上，针对现有的基于 P2P 的视频直播系统中的负载均衡算法存在的问题，提出了一种分布式负载均衡算法。该算法不仅可以协同考虑节点的处理能力和网络延时，还可以在开销较小的前提下有效的降低组播树高度和平均源到端延时。

**关键词:** P2P 视频直播 负载均衡 分布式

## 1 引言

与传统的 C/S 模式相比，基于 P2P 技术的非中心结构的对等流媒体应用模式，具有资源利用率高、网络性能好、资源标识方法统一、中转服务成本低、信息数据成本资源 “边缘化”等优势。因此，在一个动态的、异构的网络环境中，通常采用分布式树优先的方案，构建基于源的满足约束条件的应用层组播树来实现视频直播。而这种对于延时敏感的网络应用系统而言，从源节点到系统中其它各节点之间的网络延时应受到严格控制，即组播树的高度应该控制在一定范围内。但是，由于网络的异构性新节点的加入过程总是倾向于组播树上延时较小的分支，从而可能造成组播树的负载不均，进而造成媒体服务质量的下降。因此，均衡负载就成为了基于 P2P 的视频直播系统中的一个核心问题。

## 2 负载均衡算法存在的问题

通过分析，现有的负载均衡算法存在以下两点不足：(1)负载的转移没有考虑节点之间的链路延迟，所以负载转移可能在链路延迟较大的节点之间进行，浪费了系统资源；(2)算法依赖于系统中固定位置的某些节点,这些节点负责收集负载信息和生成转移策略，所以它们的失效将影响负载均衡甚至导致负载均衡不能进行。因此，为了更好的解决负载均衡的问题，更加有效的管理分布式系统的资源，进一步减小组播树高度、平均源到端延时和提高节点的带宽利用率。下面给出一种分布式负载均衡算法，该算法可以协同考虑网络各节点的处理能力，以及网络延时等问题。

## 3 分布式负载均衡算法

### 3.1 两种优化操作

(1) Degrade：对于任意节点 $v \in V$，若 $Contribution(v)=0$ 且 Level(v)较小时，即节点 V 在离根节点较近的位置上，同时对系统的贡献较小，则对节点 V 实行 Degrade 操作，使节点 V 向树叶节点方向移动，将节点 V 重定向到其兄弟节点上，如图 1 所示。

(2) Upgrade：若组播树上节点 V 对系统的贡献 $Contribution(v)$ 较大，则对节点 V 执行 Upgrade 操作，使节点向根节点方向移动，如图 2 所示。

其中，对于系统中的任意节点 $v \in V$ 定义其对系统的贡献为 $Contribution(v)$，用该节点现有的孩子节点数来表示，即 $Contribution(v) = d_{used}(v)$，显然叶节点对系统的贡献为 0；对于任意节点 $v \in V$，定义 Level(v)=Level( $N_P(v)$ )+1,( $N_P(v)$ 表示节点 V 的父节点，根节点 r 的层 Level(r)=0)表示其在组播树上的层。显然，靠近根节点的节点层次较低，叶节点的层次较高。

① 收稿时间:2009-04-01

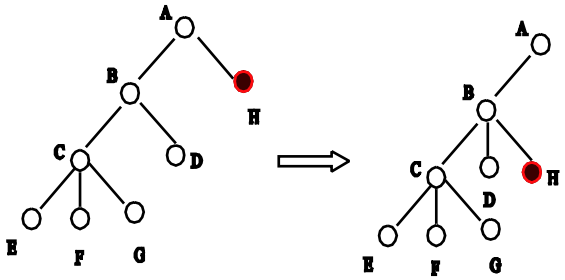


图 1　节点 H 执行 Degrade 操作

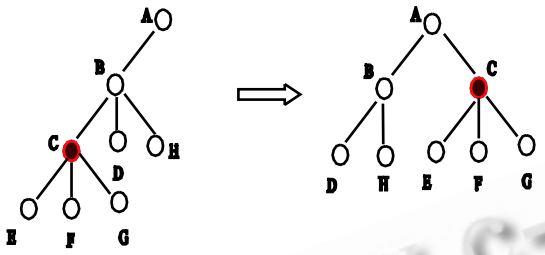


图 2　节点 C 执行 Upgrade 操作

在上面描述的 Degrade 和 Upgrade 操作中，待优化节点 V 总是通过其原父节点 $N_P(v)$ 寻找合适的新父节点，若节点 $N_P(v)$ 在局部找不到合适的节点接替自己为节点提供服务，则将该消息向 $N_P(v)$ 的子节点或父节点进行扩散，直到找到合适的节点为止；同时，在执行 Degrade 和 Upgrade 操作和节点重定向的过程中，节点 V 的原父节点 $N_P(v)$ 并不立即停止向节点 V 提供服务，而是等到节点 V 的新父节点成功接收节点 V 作为自己的孩子节点，才停止为节点 V 提供服务，以保证媒体服务的连续性。

### 3.2 算法描述

Step1：若节点 V 为根节点则直接发起负载均衡算法，进入步骤 Step2；否则,若节点 V 为普通节点，则待收到标记为 Opt 的优化消息后开始进行优化算法，进入步骤 Step2；

Step2：节点 V 首先判断自身是否已经达到饱和(即 full=true)，同时存在一个对系统贡献为 0 的子节点；如果存在这样的子节点，则向其发送标记为 Degrade 的消息并捎带子节点信息 $N_C(v)$，使其执行降级操作；该待优化节点选取兄弟节点中延时较小的分支进行重定向；

Step3：节点 V 选取子节点中对系统贡献最大的分支 x，V 向子节点 x 发送标记为 Upgrade 的消息，使其执行升级操作；节点 x 选取仍有空余度(即 full=false)的祖先节点进行重定向；若节点 v 所有的子节点对系统的贡献均为 0，则不执行任何操作；

Step4：节点 v 向其所有子节点发送标记为 Opt 的优化消息；

在节点在加入系统后，组播树正常分发视频数据的同时，自顶向下的每个节点将周期性地执行该算法。这里，虽然优化周期 T 的选择越小越好，但周期变小，算法开销将会增大，组播树也会处在一个不稳定的状态；而周期 T 的选择过大，又不能及时对过载节点实施负载均衡算法。因此，在分布式负载均衡算法中，周期 T 的选择应依据系统的规模和网络的状况而定，需要受到严格的控制。T 的选择是影响组播路由性能的关键因素，但是，由于问题的复杂性，严格的理论分析尚未获得，本文将通过仿真实验的方法对其进行研究。

### 3.3 动态适应环境

在执行负载均衡算法的过程中，不可避免系统的稳定性将受到一定程度的影响，但是由于在该视频直播系统中的任意节点都有一个独立的 Buffer 缓存最近固定时间长度内的媒体数据，因此在节点根据均衡负载算法进行负载转移的过程中，播放器仍然可以继续播放，而不会影响到视频质量和效果。

## 4　仿真实验和性能分析

### 4.1 仿真环境的建立

仿真实验是在非结构化 P2P 视频直播系统 SSAMP 上进行的，其底层网络拓扑是基于 GT-ITM 生成的。该网络拓扑采用中转和桩(TS)模型，共随机产生 1080 个路由器节点,节点的链路连接采用 Wax man2 模型(取 $\alpha=0.6, \beta=0.6$)。相邻两路由器节点间的通信延时与其几何距离成正比例，在 1ms-l00ms 之间取值(忽略路由器的转发延时)。任意两路由器节点之间的通信采用延时最短路径路由。主要实验参数可设置如下：系统节点规模(用 n 表示)最大为 1000(即 $n_{\max}=1000$),节点的度约束值在 3-6 之间随机平均分布，邻居节点间交换 Heartbeat/Refresh 报文的周期为 2s，一次仿真的运行时长为 120m，状态统计时间间隔为 10s，视频数据传输速率 r=200kbps。

### 4.2 仿真结果

#### 4.2.1 负载均衡效果

图 3 表示在静态模型下，当节点总数在较大规模(n=1000)的情况下，系统执行优化算法的次数与组播树平均高度 H 的关系。不难发现，随着优化算法的执行，组播树平均高度 H 不断减小。这从一方面反映出

优化算法对系统性能提高有一定的作用，促使组播树的直径和平均源到端延时逐渐减小，系统带宽得到更合理的利用。

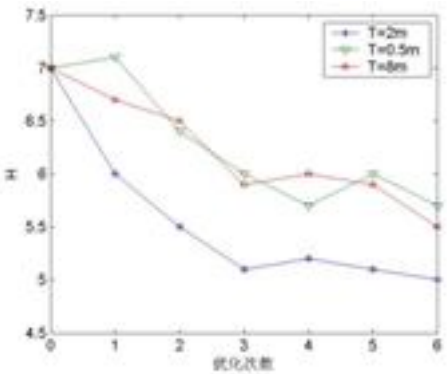


图 3 优化次数和组播树平均高度 H 关系图

下面分析优化周期 T 的取值对 H 的影响：首先，无论优化周期 T 如何选取，优化算法在不同程度上对组播树平均高度 H 的降低都有一定的作用；此外，当优化周期 T=0.5m 时，由于优化算法被频繁的执行，在一定程度上打乱了组播树的结构，较多的节点受到影响，造成媒体服务的不稳定，组播树高度也随之产生一定程度的波动；而当优化同期 T=8m 时，优化算法的效果相对较弱；相比之下，优化同期 T=2m 时取得了较理想的优化效果，组播树的高度得到有效的控制且保持了媒体服务的稳定性，并且，在优化算法开始执行的前几个周期内，组播树平均高度 H 的降低比较明显，而随着组播树的结构不断趋于合理，优化算法的效果亦不断减弱。

在同等节点规模的动态模型下，系统执行负载均衡算法取得了与静态模型类似的结果。随着优化算法的执行，多播树平均高度 H 也呈现出不断减小的趋势，只是由于节点的动态性，使得优化算法的效果较之静态模型稍有减弱，但仍取得了令用户满意的效果。

### 4.2.2 负载均衡开销

负载均衡算法在一定程度上扰乱了组播树的结构，造成一定的系统开销。图 4 对负载均衡算法的开销进行了定量统计。这里，设定负载均衡算法的开销指在算法执行过程中进行负载转移操作的节点个数。与之相比较的是节点离开操作的开销，定义为以离开节点为根的子树上的节点总数。从图 4 中可以看出，负载均衡算法的开销远远小于节点离开操作的开销，在用户可以接受的范围之内。

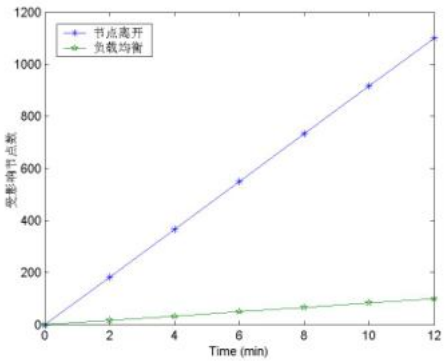


图 4 负载转移开销

## 5 结论

本文针对基于 P2P 的视频直播系统中现有负载均衡算法存在的问题，提出了一种分布式负载均衡算法。该算法不仅可以协同考虑节点的处理能力和网络延时问题，同时还具有以下优点：①可以很好的解决单点失败问题，增加了算法的鲁棒性；②负载转移在链路延时较小的分支之间进行，减少了负载均衡算法的开销；③经过几轮优化，系统中性能较差的节点因为对系统的贡献较小，被定位到组播树叶节点的位置，使组播树的结构不断趋于合理。经过理论分析和仿真实验表明，该算法可以在开销较小的前提下有效地减小组播树高度和平均源到端延时。

## 参考文献

1 Jiang XX, Dong Y, Xu DY, Bhargava B. GnuS-tream:A P2P media streaming prototype. Proc. of IEEE International Conference on Multimedia&Expo(ICME2003). 2003.6.

2 Banerjee S, Lee S, Bhattacharjee B, Srinivasan A. Resilient multicast using overlays. IEEE/ACM SIGMETRICS Performance Evaluation Review, 2003, 6(31):102－113.

3 Chawathe Y, Ratnasamy S, Breslau L, et al.Making Gnutella-like P2P systems scalable. Proc. of ACM SIGCOMM 2003. Karlsruhe, Germany, August 2003.

4 Tran D, Hua K, Sheu S. Zigzag:An efficient peer-to-peer scheme for media streaming. Proc. of IEEE INFOCOM. 2003.

5 Vahdat A, Yocum K, Walsh K, et al. Scalability and accuracy in a large-scale network emulator. ACM OSDI'02. 2002.