

P2P 分层流媒体中数据分配算法*

刘亚杰⁺, 张鹤颖, 窦文华, 陈俊峰

(国防科学技术大学 计算机学院, 湖南 长沙 410073)

Data Allocation Algorithms in Layered P2P Streaming

LIU Ya-Jie⁺, ZHANG He-Ying, DOU Wen-Hua, CHEN Jun-Feng

(School of Computer, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: Phn: +86-731-4575988, Fax: +86-731-4575988, E-mail: liuyajie@nudt.edu.cn, <http://www.nudt.edu.cn>

Liu YJ, Zhang HY, Dou WH, Chen JF. Data allocation algorithms in layered P2P streaming. *Journal of Software*, 2006,17(2):325-332. <http://www.jos.org.cn/1000-9825/17/325.htm>

Abstract: Data allocation is a key problem in layered peer-to-peer streaming with the pattern of multiple senders and single receiver. To improve the requesting nodes' streaming qualities and to reduce the bandwidth consumption of the root node, the above problem is discussed in two scenarios. The first scenario is that the root node doesn't participate in the allocation session, and the allocation goal is to maximize the requesting nodes' streaming qualities. A search and cut accurate algorithm and a heuristic approximation algorithm are presented. The second scenario is with the root node's participation, and the goal is to satisfy the request nodes' streaming qualities and to maximally save the bandwidth consumption of the root node. The problem's computing complexity in the latter scenario is analyzed and a heuristic approximation algorithm is also presented. Simulation studies show that the proposed algorithms have improved performances than the related proposed algorithms with different parameters.

Key words: P2P; layered streaming; data allocation; NP hard; algorithm

摘要: 在多对单传输模式下,数据分配是 P2P 分层流媒体中的核心问题。为了提高请求节点服务质量,同时也为了减少对 Root 节点带宽的占用,分两种情形予以讨论。一种是 Root 节点不参与的情形,其目标是最大化请求节点的服务质量。对此提出了一种基于多叉树搜索裁剪的精确算法和一种启发式近似算法。另一种是 Root 节点可参与的情形,其目标是在满足请求节点服务质量的同时,最大化节约 Root 节点的带宽资源。分析了该情形下目标问题的复杂性,提出一种启发式近似算法。仿真实验表明,在不同参数条件下,所提出的算法比同类算法都有性能上的改进。

关键词: P2P; 分层流媒体; 数据分配; NP 难; 算法

中图法分类号: TP393 文献标识码: A

P2P 流媒体服务体系下的 Peer 节点一般具有以下几个特性:1) 与传统的服务器节点相比,Peer 节点作为普通的主机节点,其所能够或愿意提供的带宽资源有限,而流媒体数据率较高,因此通常需要多个节点才能为单个

* Supported by the National Natural Science Foundation of China under Grant Nos.60433040, 90104001 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.2003CB314002 (国家重点基础研究发展规划(973))

Received 2003-11-21; Accepted 2005-07-11

节点提供数据服务;2) 不同 Peer 节点的带宽资源具有异构性,因而能接收和处理不同 QoS 等级的流数据;3) 不同 Peer 节点所缓存数据的异构性导致其对外所能提供的数据内容也不同;4) 同一 Peer 节点的上行、下行带宽也可能具有异构性.分层流媒体能够适应节点的异构性,在分层流媒体中,节点接收到的数据层次越多,服务质量越好.

本文的工作基于分层流媒体,其主要内容是在多发送方/单接收方的模式下,研究如何在不同的发送/接收对之间对数据进行优化分配,以最大化请求节点的服务质量,同时减少对源服务器节点(Root 节点)带宽的占用.这里我们把数据分配的粒度定义为数据层.在对目标问题进行求解的过程中,本文把上述优化问题分为两种情形进行讨论:一种是 Root 节点不参与的情形,即 Root 节点与请求节点之间有连接故障或者 Root 节点上已无空闲带宽资源,该情形下的数据层分配目标是最大化请求节点的服务质量;另一种是 Root 节点可参与的情形,即 Root 节点可单独为请求节点提供其所期望的部分或全部的数据层.该情形下的数据层分配目标是最大化请求节点服务质量的同时最小化占用 Root 节点的带宽资源.本文的工作不涉及如何为请求节点查找到多个 Peer 服务节点,实际上,对于 Peer 节点的搜索问题,已有很多相关工作予以探讨^[1,2],在此不作专门研究.

1 相关研究工作

在多对单传输模式下如何对流媒体数据进行分配调度是近几年的研究热点.按照分配调度粒度,可把已有工作划分为基于数据包的细粒度调度和基于数据层的粗粒度调度.在细粒度调度中,文献[3]提出了一种基于 TCP-friendly 带宽测试的数据分配算法,其目的是确保每个数据包仅被一个发送节点所发送,同时使数据包的丢失及延迟最少;文献[4]提出了一种以 FEC 编码为基础的数据分配算法,其目的是减少数据包在突发丢包网络环境下的丢失概率;文献[5]提出了一种结合网络拓扑发现的 P2P 流媒体服务体系 PROMISE,它主要考虑了多对单传输模式下可能出现的共享瓶颈网络带宽对接收方的影响;文献[6]提出了一种自适应的分层 P2P 流媒体框架,它结合网络的状态和接收方分层缓冲区的状态,引入滑动窗口机制,在各个发送方之间进行数据分配,以平滑接收方的服务质量.本文的工作与上述工作的主要区别是,本文的工作属于粗粒度调度.在粗粒度调度中,文献[7]提出了一种基于 MDC 编码的调度方法,它对每条 MDC 子流采用独立的多播树进行传输,接收方可以根据自身带宽情况选择加入到多棵多播树,从而也能适应节点的异构性.一般来讲,MDC 编码比分层编码的效率要低,且 MDC 编码的各个分层之间在解码时不存在依赖关系;文献[8]分别针对分层编码速率同构与异构这两种情况提出了两类算法:一类是在不限制数据供应节点数目前提下的数据层分配算法;另一类是在限制数据供应节点数目前提下的数据层分配算法.本文的工作仅假设异构的分层编码速率,实际上,该假设更具普遍性^[9].另外,为了在最大化请求节点服务质量的同时最小化占用 Root 节点的带宽资源,本文根据 Root 节点与请求节点之间网络带宽资源情况把目标问题划分为两种情形进行了讨论,而文献[8]只统一地提出了一种数据层分配近似算法.本文的工作属于对文献[8]工作的扩展和改进.

2 目标问题与算法描述

标记 Root 节点为 p_0 ,并假设 p_0 有充足的带宽资源.节点 q 所请求的流媒体为 s , s 的最大编码层数为 L_s , 数据层 i 的编码速率为 $r_i (i=1,2,\dots,L_s)$ 且速率不变,数据层 i 的解码依赖于所有的数据层 $k (1 \leq k < i)$. q 的下行带宽为 i_q ,对 s 期望获取的数据层数为 $m (m \leq L_s)$.

定义 1. 如果系统中某个 Peer 节点 p 包含节点 q 所请求的数据层,且 p 的可用上行带宽不小于 $\text{Min}(r_i) (i=1,2,\dots,m)$,则称 p 为 q 的可用节点.

设 q 的可用节点集合为 S_q , S_q 的大小为 n ,其成员序列为 $p_1, p_2, \dots, p_n, p_j (j=1,2,\dots,n)$ 上所缓存的 s 的数据层数为 $a_j (1 \leq a_j \leq L_s)$, p_j 的可用上行带宽为 $o_j (j=1,2,\dots,n)$.显然, p_j 为节点 q 所提供的数据层范围不能超过 a_j ,速率之和不能超过 o_j .

2.1 Root节点不参与的情形

由于 Root 节点不参与数据分配,即集合 $p_0 \notin S_q$, 因此在该情形下目标问题为如何在 S_q 集合成员节点间对 q 所请求的数据层进行分配,以使 q 的服务质量最大.此外,考虑分层流媒体中数据层之间的解码依赖关系,该问题可描述如下:

定义 2. 设有序集合 $S = \bigcup_{i=1}^m r_i$, 如果 T 为 S 的有序子集,且 T 的形式符合 $T = \bigcup_{i=1}^k r_i (1 \leq k \leq m)$, 则称 T 为 S 的前序子集,记为 $T \subset^f S$.

定义变量 $x_{i,j} \in \{0,1\} (i=1,2,\dots,m, j=1,2,\dots,n)$, $x_{i,j}=1$ 表示在数据分配过程中,数据层 i 被分配给集合 S_q 中的节点 p_j ; 否则, $x_{i,j}=0$.

对待分配的 m 层数据层,按其编码顺序从低到高排列,组成有序集合 $A = \{r_1, r_2, \dots, r_m\}$, $r_i (i=1,2,\dots,m)$ 的定义同前文.设 T 为 A 的前序子集,则本节的优化目标为

$$\text{Maximize } |T| (T \subset^f A) \tag{1}$$

同时满足如下约束:

$$x_{i,j} \leq a_j (i=1,2,\dots,m, j=1,2,\dots,n) \tag{2}$$

$$\sum_{i=1}^{|T|} x_{i,j} r_i \leq o_j (j=1,2,\dots,n) \tag{3}$$

$$\sum_{j=1}^n x_{i,j} = 1 (r_i \in T) \tag{4}$$

式(1)表示按顺序为 q 尽量多地分配数据层,即最大化请求节点的服务质量,式(2)表示节点 p_j 所提供的数据层范围不能超过其缓存的数据层范围,式(3)表示节点 p_j 所提供的速率和不能超过其可用上行带宽 o_j , 式(4)表示前序子集 T 中的每一层数据都得到分配.实际上,Root 节点不参与的数据层优化分配问题为 NP 难问题^[8].

2.1.1 基于搜索裁减的数据层分配算法 SCA-WOR(search and cut algorithm-without root)

算法的主要思想是先以 m 层数据和 S_q 中节点 $p_j (1 \leq j \leq n)$ 为参数构建一棵多叉树,再把目标问题转化为对多叉树的搜索裁剪问题.多叉树的构建方法如图 1 所示.树的高度为 m ,对树中第 $m-i+1 (1 \leq i \leq m)$ 层的每个结点, $\forall p_j \in S_q$, 如果 $a_j \geq i$ 且 $o_j \geq r_i$, 那么从第 $m-i+1$ 层所有结点上生成一个子结点,表示数据层 i 可以分配给节点 p_j , 标识连接该父子结点的边为 p_j , 依此类推生成多叉树.最后,按宽度优先顺序对所有结点进行标识,根结点标识为 1, 并给每个结点 $k (k \geq 1)$ 绑定向量 $V_k = (o'_1, o'_2, \dots, o'_n)$, $o'_j (1 \leq j \leq n)$ 代表在多叉树结点 k 处 $p_j (1 \leq j \leq n)$ 的剩余上行带宽,显然 $V_1 = (o_1, o_2, \dots, o_n)$.

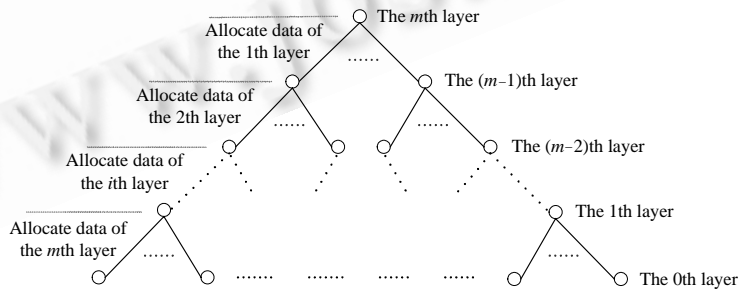


Fig.1 The multi-branch tree of data layer allocation

图 1 数据层分配多叉树

用变量 $Cut(k, l_k, p_j) \in \{\text{true}, \text{false}\} (k \geq 1, 1 \leq l_k \leq m, 1 \leq j \leq n)$ 表示是否裁剪结点 k 处以边 p_j 为连接边的子树,其中 k 表示多叉树中标识为 k 的结点, l_k 表示结点 k 在树中的层次, p_j 意义同前.如果 $V_k = (o'_1, o'_2, \dots, o'_n)$ 中的分量 $o'_j < r_{m-l_k+1}$, 则 $Cut(k, l_k, p_j)$ 取值为 true, 否则为 false.

用 $First(k)$ 和 $Last(k)$ 分别表示当前结点 k 处未搜索过的子结点的起始标识和终止标识. 搜索裁剪算法的框架如下:

SCA-WOR (search and cut algorithm of layer allocation without the root node's participation).

- (1) 置 k 为 1, 集合 U 为空;
- (2) WHILE k 大于 0
- (3) IF $First(k)$ 小于 $Last(k)$
- (4) FOR $v = First(k)$ TO $Last(k)$
- (5) 对边 kv , 取其标识 p_j ;
- (6) IF $Cut(k, l_k, p_j)$ 为 true, 裁剪 k 的以边 kv 为连接的子树, 把 k 加入到集合 U
- (7) ELSE 置 V_v 值为 V_k , 再在其分量 o'_j 上减去 r_{m-l_k+1} , 置 k 的值为 v , IF 结点 k 的层次为 0, 则把 k 加入到集合 U , 算法终止;
- (8) ENDFOR
- (9) 置 k 值为其父结点的标识, IF 结点 k 的父结点不存在, 置 k 值为 0;
- (10) END WHILE
- (11) 从集合 U 中找出层次最低的结点, 标识为 k , 输出 k .

同前文, l_k 表示结点 k 在二叉树中的层次, 则算法的输出代表能从集合 S_q 中为 q 分配前 $m-l_k$ 层编码数据, 由根结点 1 到结点 k 所组成的边序列路径代表分配方案, 即把数据层 $i (1 \leq i \leq m-l_k)$ 分配给边序列路径中第 i 条边所标识的 Peer 节点.

定理 1. SCA-WOR 算法所求出的解是 Root 节点不参与的数据层分配问题最优解.

证明: 由上文可知, 裁剪前的二叉树包括了所有可能的解, 而对二叉树的搜索裁剪结果则保留了所有满足问题约束的解, 其中层次最低的结点代表问题的最优解.

2.1.2 基于启发的数据层分配算法 HA-WOR (heuristic algorithm-without root)

算法 SCA-WOR 虽然可以获得最优解, 但算法的时间复杂性也比较高, 最坏情况下达到 $O(n^{m+1})$. 本节提出一种基于启发的数据层分配算法, 其主要思想包括: 1) 按编码顺序从低到高依次对数据层进行分配; 2) 在对每层数据 $i (i=1, 2, \dots, m)$ 进行分配时, 选择 S_q 集中所有符合限制条件的节点 (假设节点组成的集合为 S'_q); 3) 把数据层 $i (i=1, 2, \dots, m)$ 优先分配给 S'_q 中剩余上行带宽最小, 同等条件下其缓存的数据层数目又最少的节点. 算法框架如下文所示, 算法的时间复杂性为 $O(mn)$.

HA-WOR (heuristic layer allocation algorithm without the root node's participation).

- (1) 对 m 层数据层按其编码顺序由低到高排列, 得到集合 $L = \{1, 2, \dots, m\}$, 其成员代表分层编码所在的层次;
- (2) FOR $k=1$ TO m
- (3) 从序列 L 中取出第 k 个成员, 设其对应的数据层为 i , 置集合 S'_q 为空;
- (4) FOR $j=1$ TO n
- (5) 对节点 p_j , 如果 $a_j \geq i$ 且其剩余上行带宽大于 r_i , 则把节点 p_j 加入到集合 S'_q 中;
- (6) END FOR
- (7) 如果 S'_q 为空, 表示只能对前 $i-1$ 层数据进行分配, 则算法退出;
- (8) 选取集合 S'_q 中剩余上行带宽最小的节点; 如果选出的节点有多个, 则从中选取缓存数据层数目最少的节点; 把数据层 i 分配给所选出的节点, 更新该节点的剩余上行带宽;
- (9) END FOR

2.2 Root 节点可参与的情形

首先考虑该情形下的一种特殊情况, 即 Root 节点 p_0 有充足的带宽资源, 能为 q 提供任何数据层. 由于仅 p_0 就能满足 q 的服务质量, 而在分配过程中为了减少对 Root 节点带宽的占用, 应先把数据层尽量分配给 S_q 中的

Peer 节点,而仅仅把不能分配给 S_q 的剩余数据层再分配给 p_0 . 因此,在该特殊情形下的目标问题可以转换为:如何在 $p_j (j=0,1,\dots,n)$ 之间对数据层进行分配,在满足 q 对 m 层数据的请求的前提下,使得 p_0 被占用的带宽资源最少. 该优化问题的目标及约束可描述如下:

优化分配目标:

$$\text{Maximize } \sum_{i=1}^m \sum_{j=1}^n x_{i,j} r_i \quad (5)$$

并有如下约束:

$$x_{i,j} i \leq a_j (i=1,2,\dots,m, j=1,2,\dots,n) \quad (6)$$

$$\sum_{i=1}^m x_{i,j} r_i \leq o_j (j=1,2,\dots,n) \quad (7)$$

$$\sum_{j=0}^n x_{i,j} = 1 (i=1,2,\dots,m) \quad (8)$$

定理 2. 对给定异构数据层集合 $L = \{1,2,\dots,m\}$ 和节点集合 S_q , 在满足约束式(6)和式(7)的前提下判定 L 中的数据层是否能够完全被分配到集合 S_q 中的问题属于 NP 完全问题.

证明:把集合 L 中的每个数据层 i 映射为一个物体,该物体的体积对应于数据层 i 的速率 r_i ; 把 S_q 集合中的每个 Peer 节点 p_j 映射为一个背包,该背包的容量为 o_j ; 把 p_j 上所缓存数据层的限制映射为对每个背包,只允许一些特殊指定的物体放入其中. 这样,上述判定问题与带约束的多背包问题等价,而带约束的多背包问题属于 NP 完全问题^[10]. 另外,对于给定的某个分配候选解,显然可以在多项式时间内验证该候选解是否正确.

由定理 2 可知,在满足约束式(6)和式(7)的前提下求解(5)的优化分配问题是 NP 难问题.

2.2.1 基于启发的数据层分配算法

针对上述优化问题,本节提出一种新的启发式算法 HA-WR(heuristic algorithm-with root),其思想与第 2.1.2 节中的 HA-WOR 算法实现类似,所不同之处是,在分配过程中对数据层按其编码速率的大小,从高到低依次分配. 算法框架如下文所示,该算法的时间复杂性为 $O(mn)$.

HA-WR (heuristic layer allocation algorithm with the root node's participation).

- (1) 对 m 层数据层按其编码速率大小非递增排序,设排序结果为集合序列 L ;
- (2) FOR $k=1$ TO m
- (3) 从序列 L 中取出第 k 个成员,设其对应的数据层为 i ,置集合 S'_q 为空;
- (4) FOR $j=1$ TO n
- (5) 对节点 p_j ,如果 $a_j \geq i$ 且其剩余上行带宽大于 r_i ,则节点 p_j 加入 S'_q ;
- (6) ENDFOR
- (7) IF S'_q 为空,则把数据层 i 分配给 Root 节点 p_0 ,CONTINUE;
- (8) ELSE 选取集合 S'_q 中剩余上行带宽最小的节点;如果选出的节点有多个,则从中选取缓存数据层数目最少的节点,把数据层 i 分配给所选出的节点,更新该节点的剩余上行带宽;
- (9) ENDFOR

考虑 Root 节点可参与情形下的其他情况,如 Root 节点 p_0 和请求节点 q 之间的可用带宽仅仅允许分配传送 L 中的部分数据层,而不能分配传送 L 中的全部数据层. 为了最大化节点 q 的服务质量,同时兼顾最小化占用 p_0 的带宽资源,在 Root 节点可参与的情形下,一般算法的主要步骤如下:先运行上述 HA-WR 算法,检查 p_0 和 q 之间的可用带宽是否足够传输那些不能分配到节点集合 S_q 的剩余数据层,如果足够,则算法结束;否则,把 p_0 加入到集合 S_q 中,并运行第 2.1.1 节中的 SCA-WOR 算法或第 2.1.2 节中的 HA-WOR 算法.

3 仿真实验与结果分析

3.1 仿真实验环境

在仿真实验中,我们首先用工具 GT-ITM^[11]生成一个具有两个层次的网络拓扑,分别代表核心路由层和边缘路由层,核心路由层由 1 个域组成,它包含 4 个核心路由节点;边缘路由层由 12 个域组成,每 3 个域附属于一个核心路由节点,边缘路由层的每个域中平均包含 8 个边缘路由节点,每个边缘路由节点代表一个网络区域,网络拓扑中的带宽定义如下:核心路由节点之间、核心路由节点和边缘路由节点之间的带宽为 1 000M,边缘路由节点之间的带宽为 100M,同属一个网络区域的主机节点之间的网络带宽仅受两个主机节点之间接入带宽的限制,节点间的路由路径由最短路径算法 Dijkstra 决定。

其次,生成 40 000 个节点并随机附加到边缘路由层的 96 个路由器上,这些节点按带宽的属性可划分为 3 类:第 1 类是 Modem/ISDN 节点,占节点总数的 50%,其主机接入带宽为 112Kbps;第 2 类是 Cable Modem/DSL 节点,占节点总数的 35%,其主机接入带宽为 1Mbps;第 3 类节点是 Ethernet 节点,占节点总数的 15%,其主机接入带宽为 10Mbps.把 Root 节点直接附加到 4 个核心路由层中的某个节点上,其主机接入带宽为 100Mbps.而对于两个节点之间的可用带宽不仅受其主机接入带宽的限制,而且受二者所附接路由器之间最短路径上的可用网络带宽的限制.目标节目的播放时长为 60 分钟.假设目标节目被分层编码后,其最低层的速率 r_1 为 20Kbps,第 i 层的速率 r_i 为 $(i-1) * r_1$ ($i = 2, 3, \dots, 10$),编码速率之和为 920Kbps.

最后,在每次实验过程中模拟 VoD 点播过程,从 40 000 个节点中依次随机选取节点加入系统请求服务,加入过程服从参数为 λ 的泊松分布, λ 的取值作为本次实验参数来指定.每个节点均从节目的初始位置开始请求数据,节点加入系统后缓存本次实验指定时长的其最近所接收到的数据,从而可作为后续该时长内加入系统的其他节点的候选服务节点.节点在加入后 60 分钟内不离开.每次实验中的 VoD 系统均运行 24 小时,而为了在实验中制造 Root 不参与的情形,假设路由器之间有 98% 的网络资源已被其他业务所占用.由于仿真实验包含了节点缓存时间、节点上行/下行带宽比率以及节点平均请求到达率 λ 等实验参数,因此在实验中我们分别对上述参数条件下的系统总体性能进行了评估.对于每种实验参数组合,实验共进行 10 次,结果取平均值.

3.2 仿真结果分析

为了衡量节点的服务质量情况,这里我们定义了节点质量满意度^[8],它为节点实际所接收的数据层数与其所期望的数据层数的比率,满意度越大表明节点的服务质量越好.命名文献[8]所提出的相应算法为 OHA(original heuristic algorithm),实验结果也主要与该算法的结果进行比较.图 2 给出了系统节点的平均质量满意度,其中的图 2(a)~图 2(c)都分别对应着不同的节点缓存时间、不同的节点上行/下行带宽比率、不同的节点请求到达率等实验条件.从图 2 可以看出,对于典型的 VoD 服务过程,在不同的实验参数条件下,使用本文所提出的算法在一般情况下均能使节点的满意度得到不同程度的提高.图 2(a)的结果表明,系统总体质量满意度随节点缓存长度的增加而有所提高,这是因为节点缓存越长,节点在加入时能为其提供数据服务的 Peer 可用节点平均数越多,从而能够使系统总体的质量满意度提高.至于上行/下行带宽比率对质量满意度的影响,从图 2(b)可以看出,上行/下行带宽比率越大,系统总体质量满意度也就越高,这是因为上行/下行带宽比率越大,Peer 节点可为请求节点提供的上行带宽也越大.从图 2(c)可以看出,系统总体满意度随 λ 值的增加而降低,通过对实验过程进行统计,我们发现,当 λ 很小时,加入节点所请求的数据层基本上都由 Root 节点所提供,且由于实验过程中加入节点的数目也较小,因此满意度基本上可达 1;而随着 λ 值的增加,虽然节点加入时 Peer 可用节点的平均数目也随之增加,但路由器之间的空闲带宽资源也随之减少.另外,由于上行/下行带宽比率小于 1,使得新节点加入时不能够分配获取全部期望数据层的节点数与系统中节点总数的比例也越大,从而导致系统质量满意度随 λ 值的增加而降低.另外,算法 SCA-WOR 和 HA-WR 的联合比算法 HA-WOR 和 HA-WR 的联合对系统质量满意度的提高不明显,通过对实验中的不同情形进行统计发现,Root 节点不参与的情形在实验中出现的频率不高,从而使前者与后者相比,对整个系统总体性能的改进不明显.

图 3 给出了 Root 节点带宽资源被占用的情况,其中图 3(a)~图 3(c)也分别对应着不同的节点缓存时间、不

同的节点上行/下行带宽比率、不同的节点请求到达率等实验参数.从图 3 可以看出,本文所提出的算法在节约 Root 节点带宽资源的占用方面也优于文献[8]所提出的相关算法.图 3(a)的结果表明,Root 节点带宽资源的占用随着节点缓存长度的增加而减少,这是因为在其他实验条件不变的前提下,随着节点缓存时长的增加,Peer 节点的上行带宽也被其他节点利用得更好,从而减少了对 Root 节点带宽资源的占用.对于上行/下行带宽比率设置的影响,从图 3(b)可以看出,上行/下行带宽比率越大,Root 节点被占用的带宽资源越少,这是因为上行/下行带宽比率越大,Peer 节点可为请求节点提供的带宽资源也越多.从图 3(c)可以看出,Root 节点带宽资源的占用随 λ 值的增加而增加,这是因为上行/下行带宽比率小于 1,而随着 λ 值的增加,单位时间内请求到达系统的节点数目越多,Root 节点的带宽资源也被占用得越多.算法 SCA-WOR 和 HA-WR 的联合比算法 HA-WOR 和 HA-WR 的联合对 Root 节点带宽资源占用有少量的减少,这是因为在 Root 节点不参与的情形下,前者能为请求节点分配更多的数据层,在同等条件下,这些请求节点又可能为其他候选节点提供更多的数据层,从而能够减少对 Root 节点带宽资源的占用,同时,在每次实验过程中,这种情形发生的实例次数相对不多.

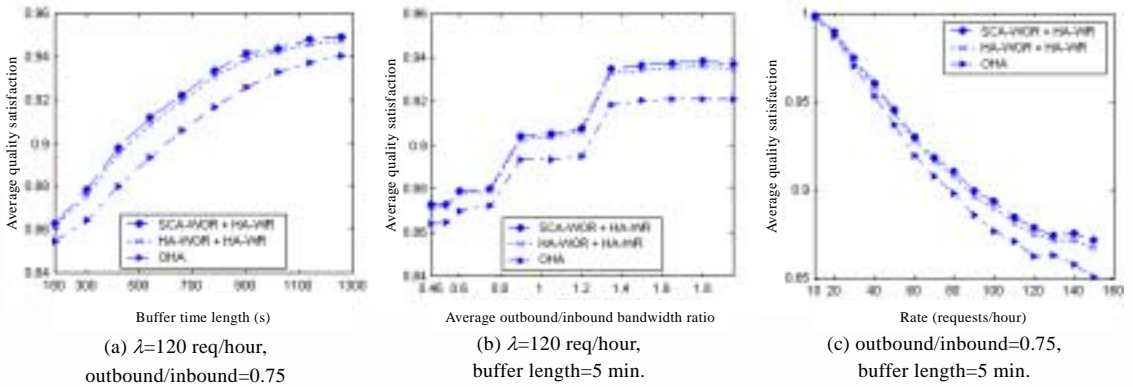


Fig.2 Overall streaming quality with different experiment parameters

图 2 不同实验参数条件下的节点质量满意度

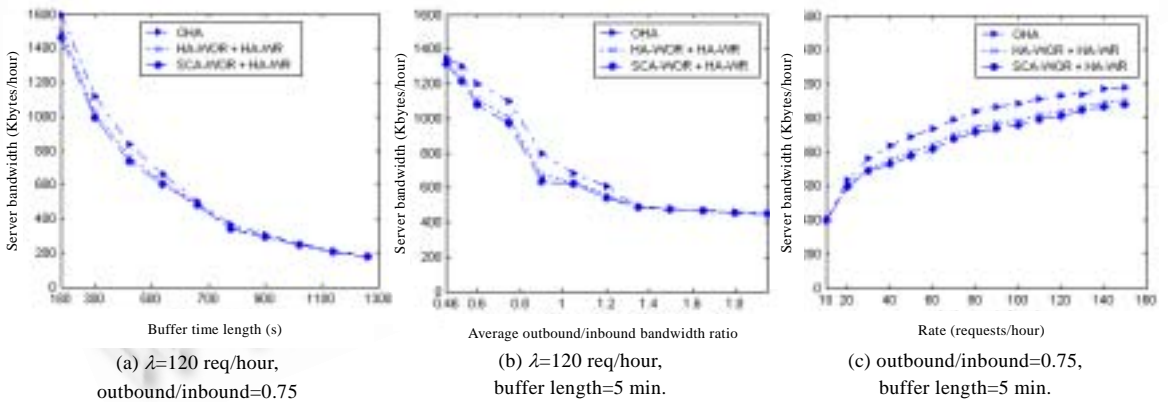


Fig.3 Root node bandwidth consumption with different experiment parameters

图 3 不同实验参数情况下的 Root 节点带宽占用情况

此外,在仿真实验中我们对另外一种分层编码模式(其编码速率从低层到高层依次分别为{20,20,96,64,32,128,160,192,128,64}Kpbs)的情形下节点质量满意度以及 Root 节点带宽资源的占用情况进行了评估,其结果与前一种编码模式具有相似的特征,故没有在此列出.

4 结束语

分层流媒体能够适应 Peer 节点资源能力的异构性.本文研究了 P2P 分层流媒体中基于数据层的分配问题,其目标是在最大化请求节点服务质量的同时,尽量减少对 Root 节点带宽资源的占用.根据 Root 节点与请求节点之间带宽资源的情况,我们分两种情形对目标问题进行了讨论,分析了目标问题的复杂性,并提出了相关算法.通过对一种 P2P VoD 系统进行 24 小时的模拟运行,其结果表明,本文所提出的算法较之相关算法既能够提高系统节点总体服务质量,同时也能够减少对 Root 节点带宽资源的占用.

References:

- [1] Stephanos AT, Diomidis S. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys*, 2004,36(4): 335-371.
- [2] Lua EK, Crowcroft J, Pias M, Sharma R, Lim S. A survey and comparison of peer-to-peer overlay network schemes. *Journal of IEEE Communications Survey and Tutorial*, 2005,7(2).
- [3] Nguyen T, Zakhor A. Distributed video streaming over the Internet. In: Kienzle MG, ed. *Proc. of the SPIE Conf. on Multimedia Computing and Networking 2002*. Bellingham: SPIE Press, 2002.
- [4] Nguyen T, Zakhor A. Distributed video streaming with forward error correction. In: Hemy M, ed. *Proc. of the Packet Video Workshop*. New York: IEEE Press, 2002.
- [5] Hefeeda M, Habib A, Botev B, Xu D, Bhargava DB. PROMISE: A peer-to-peer media streaming system. In: Lawrence AR, ed. *Proc. of the ACM Multimedia 2003*. New York: ACM Press, 2003.
- [6] Rejaie R, Ortega A. PALS: Peer-to-Peer adaptive layered streaming. In: Christos P, Kevin CA, eds. *Proc. of the ACM NOSSDAV 2003*. New York: ACM Press, 2003.
- [7] Padmanabhan VN, Wang HJ, Chou PA, Sripanidkulchai K. Distributing streaming media content using cooperative networking. In: Kevin A, ed. *Proc. of the ACM/IEEE NOSSDAV 2002*. New York: ACM Press, 2002.
- [8] Cui Y, Nahrstedt K. Layered peer-to-peer streaming. In: Christos P, ed. *Proc. of the ACM NOSSDAV 2003*. New York: ACM Press, 2003.
- [9] Kim T, Ammar M. A comparison of layering and stream replication video multicast Scheme. In: Jason N, ed. *Proc. of the ACM NOSSDAV 2001*. New York: ACM Press, 2001.
- [10] Hochbaum D. *Approximation Algorithm for NP-Hard Problem*. Beijing: PWS Publishing Company, 1998.
- [11] Calvert K, Doar M, Zegura E. Modeling Internet topology. *IEEE Communication Magazine*, 1997,35(6):160-163.



刘亚杰(1975 -),男,湖南桃源人,博士生,主要研究领域为 P2P 网络,流媒体传输调度.



颜文华(1946 -),男,教授,博士生导师,主要研究领域为计算机网络.



张鹤颖(1976 -),女,博士,助理研究员,主要研究领域为拥塞控制.



陈俊峰(1978 -),博士生,主要研究领域为无线网络流媒.