基于批处理传输方案的移动流媒体系统的缓存分配算法

雷正雄 廖建新 朱晓民
(北京邮电大学网络与交换技术国家重点实验室 北京 100876)

摘 要：该文介绍了在 WCDMA 网络中基于代理的移动流媒体系统以及评价其中代理服务器缓存分配算法性能的平均网络传输成本和移动终端的平均播放启动延时这两个指标；推导出在移动批处理(MBatching)传输方案下与这些指标相对应的节省值和综合节省值的计算公式；提出了适用于移动流媒体系统的，使所有流媒体节目的总的综合节省值最大的缓存分配算法。仿真结果表明，该算法与其他分配算法相比，可使总的综合节省值更大，节省更多的网络传输成本，取得更大的字节命中率。

关键词：移动流媒体系统；移动批处理传输方案；缓存分配算法；综合节省值
中图分类号：TN915.5 **文献标识码**：A **文章编号**：1009-5896(2007)04-0906-05

Cache Allocation Algorithm in Batching Stream Transmission Scheme Based Mobile Streaming Media System

Lei Zheng-xiong Liao Jian-xin Zhu Xiao-min
(*State Key Laboratory of Networking and Switching Technology*, *Beijing University of Posts and Telecommunications*, *Beijing* 100876, *China*)

Abstract: In this paper, a proxy-based mobile streaming media system in WCDMA network is introduced. Two metrics, mean network transferring cost and mean playback startup latency of mobile terminals, to evaluate performance of a cache allocation algorithm in proxy are presented. Expressions for corresponding saving and integrated saving of the metrics under Mobile Batcing (MBatching) stream transmission scheme are developed. A cache allocation algorithm that applies to the mobile streaming media system is put forward to make the total integrated saving of all the streaming media program maximal. Simulation results demonstrate that the algorithm can make the total integrated saving larger, save more network transferring cost and achieve higher byte-hit rate than another allocation algorithm.

Key words: Mobile streaming media system; Mobile batching transmission scheme; Cache allocation algorithm; Integrated saving

1 引言

随着固定网如因特网上流媒体业务的飞速发展和广泛应用，移动流媒体业务正受到越来越多的重视[1]，更有人预言，它将是第三代移动通信(3G)中的重点业务。因此，非常有必要对移动流媒体系统进行深入研究。

缓存分配算法是移动流媒体系统的关键算法，用于在系统启动时，决定在代理服务器的有限大小的缓存空间中保存哪些流媒体节目的哪些数据。

目前，对缓存分配算法的研究并不多见。Sen等[2]提出，缓存流媒体的前缀可以更有效地减少流媒体播放的延时和抖动；Zhang等[3]提出了视频分段传输(staging)算法，通过在代理服务器缓存中存储数据率变化较大的流媒体数据，以减少流媒体服务器与代理服务器间的带宽变化，Ramesh等[4]在网络支持多播的条件下提出了使得流媒体服务器与代理服务器间的传输成本最小的缓存分配算法。Kangasharju等[5]针对分层可扩展性编码的流媒体，以层为缓存粒度，用最优化方法求解缓存分配的问题，在缓存大小有限以及服务器与代理间带宽有限的条件下最大化服务提供者的收益。Wang等[6]提出了一种使缓存的所有前缀所节省的网络传输成本最大的缓存分配算法，并设计了当网络仅支持单播时的流传输策略，还推导出了在此策略下网络平均传输成本的计算公式。

当运用到移动流媒体系统时，以上算法都存在如下的一个或多个缺点：未综合考虑减少播放启动延时；未考虑移动流媒体系统的网络只支持单播；未考虑移动终端的存储空间有限；未考虑非分层可扩展性编码的流媒体，因此并不完全适用于移动流媒体系统。

到目前为止，在已有的文献中，还没有提出过针对移动

2005-06-17 收到，2006-01-23 改回
高等学校博士学科点专项科研基金(20030013006)，国家移动通信产品研究开发专项基金项目(下一代移动智能网络的开发及应用)；电子信息产业发展基金重点项目(下一代网络核心业务平台)，电子信息产业发展基金项目(移动通信增值服务平台及应用系统)和国家高技术产业化信息化装备专项项目(支持数据增值业务的移动智能网系统)资助课题

流媒体系统特殊网络环境的缓存分配算法，本文将提出一种适用于移动流媒体系统的使所有流媒体节目的网络传输成本和播放启动延时的总的综合节省值最大的算法并分析其性能。

本文后续章节安排如下：第 2 节简单介绍移动流媒体系统的网络结构，第 3 节介绍移动流媒体系统的缓存分配算法，第 4 节对算法进行性能分析，第 5 节小结全文。

2　移动流媒体系统网络结构

一个基于 WCDMA 网络的移动流媒体系统如图 1 所示。

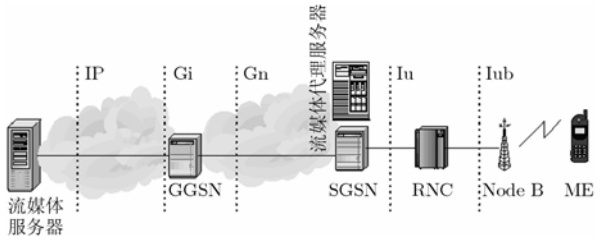

图 1 基于 WCDMA 的移动流媒体系统

系统由流媒体服务器(Streaming media server)，GGSN，SGSN，流媒体代理服务器(Proxy server)，无线网络控制器(RNC)，基站(Node B)和移动终端(ME)等模块组成。

在 WCDMA 的 R5 版本中，图 1 中的各接口均支持 IP 协议且仅支持单播，移动终端可视为移动主机，具有自己的 IP 地址，可依此被寻址并接收 IP 数据。为了使用流媒体业务，移动终端需安装流媒体播放器，并需有一定的播放节目所需的缓冲存储空间。

移动流媒体系统具有如下 3 个显著特征：核心网只支持单播；连接移动终端的无线信道只支持单播；移动终端的存储空间非常有限。

流媒体服务器一般处于因特网中，用于存放流媒体文件，流媒体代理服务器与 SGSN 放置在一起并共用同一 IP 地址。流媒体节目的数据率较大，可达 1.5Mbps，WCDMA 网络的增强技术 HSDPA(High Speed Downlink Packet Access)可满足此数据率需求，本文的讨论也假定系统可对其支持。

在系统启动时，需要使用某种规则决定保存哪些流媒体节目的哪些数据在代理服务器有限的缓存空间中。这个规则就是本文将要讨论的流媒体的缓存分配算法。

3　移动流媒体系统的缓存分配算法

3.1　缓存分配算法的性能目标

已有的缓存分配算法单独使用如下指标之一评价性能：缓存的字节命中率(BHR)或所需带宽的稳定性或网络传输成本。但这些指标并不是最有效的，而且单独使用某个指标也不能有效评价移动流媒体系统中缓存分配算法的性能。

在代理服务器的缓存中保存的前缀，对提高移动流媒体系统的性能主要有如下贡献：

首先，前缀直接从代理服务器向移动终端传输，不需从中心服务器获取，节省了整个系统的网络传输成本。

其次，移动终端直接从代理服务器获取前缀播放，其网络传输延时远小于从中心服务器获取所需延时，甚至可忽略不计，减少了移动终端的播放启动延时。

可见，对移动流媒体系统而言，节省网络传输成本和播放启动延时是代理服务器最直接的作用和目标，因此也是缓存分配算法的性能评价的基本指标。缓存分配算法的性能目标应为使得网络传输成本和播放启动延时的节省值达到最大。

为了方便比较和讨论，设节省值为与系统中无代理服务器且仅支持单播时相比较所减少的传输成本和启动延时，下节将推导出这两个节省值的计算公式。

3.2　传输成本和启动延时的节省值

假设流媒体服务器(server)上有n个恒定比特率(CBR)的流媒体，第i个流媒体节目的长度为L_i(单位为s)，数据率为b_i(单位为bps)，在代理服务器(proxy)中缓存的前缀长度为v_i(单位为s)，访问概率为p_i，代理服务器的缓存容量为S(单位为bit)，所有移动终端(client)所能支持的最小存储空间的大小为m(单位为bit)，最多可同时从两个信道接收视频流。假定对流媒体的请求到达符合泊松过程，总的请求速率为λ，第I 个流媒体的平均请求速率为λ_i，且每个请求都对流媒体从头到尾完整播放。假定在server到 proxy之间每传输流媒体 1bit消耗的网络传输成本为c_s，在proxy与client之间消耗的网络传输成本为c_p。设当系统中无代理服务器且仅支持单播时，移动终端的播放启动延时为delay(单位为s)，假定从proxy到client的网络传输延时可忽略不计。

网络传输成本随传输方案的不同而变化，为了有效地节省传输成本，结合移动流媒体系统的特征，本文提出了适合于移动流媒体系统的批处理(MBatching)传输方案。

MBatching 方案在保证满足移动终端可以连续播放不会中途中断，以及移动终端的缓冲存储空间充满流媒体数据后不用继续保存数据以至于丢失数据这两个条件下，尽可能迟地从 server 向 proxy 发送流媒体后缀，以便于批处理在某个时间段内到达的对流媒体 i 的所有请求，对这些请求，只需从 server 发送一次流媒体 i 的后缀，以达到节省网络传输带宽的目的。

如果$m/b_i>v_i$，将对流媒体i的某请求到达时刻为起始时刻的一个长度为v_i的时间段定义为一个v_i域，当前请求所在的v_i域称为当前v_i域。当前v_i域的起始时刻记为T_{v_i}。当一个新的请求于时刻t到达时，如果$t-T_{v_i}<v_i$，则请求属于当前v_i域。否则，该请求不属于当前v_i域，这时，以时刻t为起始时刻开辟一个新的v_i域，即$T_{v_i}=t$。依此类推，将不同时间到达的请求分属不同的v_i域，这样，对于流媒体i，整个时间轴被划分为以v_i域组成的划分。

对任意一个v_i域内到达的请求，proxy立即向发出请求的移动终端单播传输前缀，在该域的终止时刻，即$T_{v_i}+v_i$时刻，

才开始用一个单播流，从server取得后缀，经proxy中继后，向所有发出请求的移动终端传输。移动终端可同时从两个信道接收视频流，因此移动终端在接收前缀播放的同时可接收并存储后缀，待前缀播放完毕后播放。

在一个v_i域内，对流媒体i的请求的平均数目是$1+v_i\lambda_i$。这些请求只是从server到proxy间产生了一次后缀$[v_i,L_i]$的单播流，对每一个请求，在proxy到client间都产生了一个$[0,L]$的单播流。由于单位时间内对流媒体i的平均请求数目为λ_i，因此当$m/b_i>v_i$时，流媒体i的单位时间的平均网络传输成本：

$$C_i=\left(\frac{c_s(L_i-v_i)}{1+v_i\lambda_i}+c_pL_i\right)\lambda_ib_i \tag{1}$$

如果$m/b_i<v_i$，将对流媒体i的某请求到达时刻为起始时刻的一个长度为m/b_i的时间段定义为一个m域。与$m/b_i>v_i$时类似，对于某流媒体，根据请求的到达时刻将整个时间轴划分为以m域组成的划分。当前m域的起始时刻记为T_m。对任意一个m域内到达的请求，proxy立即向发出请求的移动终端单播传输前缀中区间$[0,\ m/b_i]$的流媒体数据，在该域的终止时刻，即T_m+m/b_i时刻，proxy向所有终端发送前缀中区间$[m/b_i,v_i]$的流媒体数据，在T_m+v_i时刻，proxy从server用一个单播流，取得后缀并发送给所有的移动终端。

在一个m域内，对流媒体i的请求的平均数目是$1+\lambda_im/b_i$。这些请求只是从server到proxy间产生了一次后缀$[v_i,L_i]$的单播流，对每一个请求，在proxy到client间都产生了一个$[0,L_i]$的单播流。因此当$m/b_i<v_i$时，流媒体i的单位时间的平均网络传输成本：

$$C_i=\left(\frac{c_s(L_i-v_i)}{1+\lambda_im/b_i}+c_pL_i\right)\lambda_ib_i \tag{2}$$

当系统中无代理服务器且仅支持单播时，对每个请求，都要从流媒体服务器启动一个单播流传输全部流媒体，流媒体i的单位时间的平均网络传输成本

$$C_i=(c_s+c_p)L_i\lambda_ib_i \tag{3}$$

因此对流媒体i，单位时间的平均网络传输成本的节省值：

$$Rc_i=\begin{cases}\left(c_sL_i-\dfrac{c_s(L_i-v_i)}{1+v_i\lambda_i}\right)\lambda_ib_i, & m/b_i>v_i\\ \left(c_sL_i-\dfrac{c_s(L_i-v_i)}{1+\lambda_im/b_i}\right)\lambda_ib_i, & m/b_i\le v_i\end{cases} \tag{4}$$

移动终端对流媒体i的播放启动延时D_i与其前缀长度有关，当前缀v_i长度大于阈值D_{th}时，D_i为0，否则，D_i等于$\text{delay}(D_{\text{th}}-v_i)/D_{\text{th}}$。而当系统仅支持单播且无代理服务器时，网络传输延时保持为delay，由于单位时间内对流媒体i的平均请求数目为λ_i，因此对流媒体i，单位时间内播放启动延时的平均节省值：

$$Rd_i=\begin{cases}(\lambda_iv_i/D_{\text{th}})\times\text{delay}, & v_i<D_{\text{th}}\\ \lambda_i\text{delay}, & v_i\ge D_{\text{th}}\end{cases} \tag{5}$$

为了综合考虑传输成本节省值和启动延时节省值，使缓存分配算法的综合性能最优，对流媒体i，定义其单位时间内的综合节省值

$$\text{ISaving}(v_i)=p_cRc_i/(c_sLb\lambda_m)+p_dRd_i/(\lambda_m\text{delay}) \tag{6}$$

其中$L=\max_{1\le i\le n}L_i$，$b=\max_{1\le i\le n}b_i$，$\lambda_m=\max_{1\le i\le n}\lambda_i$。式(6)右边的两个节省值$Rc_i$和$Rd_i$分别除以$c_sLb\lambda_m$和$\lambda_m\text{delay}$是为了使得相除的结果为无单位的值且在[0,1]区间内，以方便对它们加权求和。p_c和p_d是网络传输节省值和启动延时节省值对综合节省值的贡献权值，可用于调节两种节省值的相对比重。显然，综合节省值是v_i的函数。

对流媒体i，定义与式(6)相对应的综合考虑了传输成本和启动延时的用于性能评价的综合指标IIndex(Integrated Index)，用以在后文中评价算法的性能，IIndex定义如下：

$$\text{IIndex}(i)=p_cC_i/(c_sLb\lambda_m)+p_d\lambda_iD_i/(\lambda_m\text{delay}) \tag{7}$$

3.3 移动流媒体系统的缓存分配算法

如前所述，缓存分配算法的目的就是为了使代理服务器中缓存的所有节目的前缀的综合节省值之和达到最大。而对流媒体i，综合节省值如式(6)所示，为前缀长度v_i的函数。又因代理服务器的缓存容量S是有限的，只能容纳一部分节目的不定大小的前缀。因此缓存分配问题就是一个给定了缓存容量的约束条件后，求前缀长度v_i的最佳分配策略，使得所有节目的综合节省值之和最大的最优化问题，用公式表示为

$$\max f=\sum_{i=1}^{n}\text{ISaving}(v_i),\quad \text{s.t.}\ \sum_{i=1}^{n}v_ib_i\le S,\ 0\le v_i\le L_i \tag{8}$$

为方便求解，本文设代理服务器的缓存空间以段为单位进行分配，因此各流媒体节目的前缀大小为段的整数倍。设段大小为seg(单位为bit)，设$N=S\text{seg}$，$h_i=v_ib_i\text{seg}$，$g_i=L_ib_i\text{seg}$，因此$\text{ISaving}(v_i)$可记为$\text{ISaving}(\text{seg}\times h_i/b_i)$，由于对流媒体节目$i$，seg和$b_i$均为常数，所以可进一步记为$\text{GSaving}(h_i)$，则式(8)可记为：

$$\max f=\sum_{i=1}^{n}\text{GSaving}(h_i),\quad \text{s.t.}\ \sum_{i=1}^{n}h_i\le N,\ \ h_i\text{为整数且}\ 0\le h_i\le g_i \tag{9}$$

显然这是一个背包问题，可用动态规划法求解。动态规划法的求解分为6步，详细如下：

(1) 划分为n个阶段　将流媒体节目按序号$1,2,\cdots,n$排序，每个阶段决定1个节目在代理服务器中存储的前缀的段数。

(2) 确定状态变量　S_k表示前k个节目所允许的在代理服务器中缓存的前缀的段数之和。

(3) 确定决策变量　h_k表示第k个节目所缓存的前缀的段数。

(4) 状态转移方程为　$S_k=S_{k-1}+h_k$，允许决策集合为

$$D(S_k)=\{h_k\mid 0\le h_k\le S_k,h_k\le g_k\}$$

(5) 建立递归方程

$$\left.\begin{aligned} f_0(w)&=0, && w\in Z\\ f_k(S_k)&=\max_{h_k\in D(S_k)}\{\mathrm{GSaving}(h_k)\\ &\quad +f_{k-1}(S_k-h_k)\}, && 1\le k\le n \end{aligned}\right\} \tag{10}$$

$f_k(S_k)$表示当占用的缓存不超过S_k段时，前k个节目的综合节省值之和的最大值。

(6) 递推求解　逐步计算出 $f_1(S_1),f_2(S_2),\cdots,f_n(S_n)$，最后求得的 $f_n(N)$ 就是最大的综合节省值，相应的最优分配策略由反推运算即可求得。

为了使用计算机程序求解，我们在程序中声明了一个大小为$(n+1)\times(N+1)$的二维数组 B，$B[i][j]$代表在大小为 j 的缓存中，前 i 个节目的最大的综合节省值。按如下所示的递推关系，先按序计算出第 1 行的各列元素，然后计算第 2 行，直到计算出第 n 行的元素。

$$\left.\begin{aligned} B[i][j]&=0, && i=0\\ B[i][j]&=\max_{h_i\in D(S_i)}\{B[i-1][j-h_i]\\ &\quad +\mathrm{GSaving}(h_i)\}, && i>0 \end{aligned}\right\} \tag{11}$$

$B[n][N]$是最大的综合节省值，从此处往前反推，可以得到用以计算出这个最大值的第$n-1$ 行的元素，一直反推到第 1 行可得相应的第 1 行的元素，设相应的第i行的元素为$B[i][j]$，第$i+1$ 行的元素为$B[i][j_{i+1}]$，则第$i+1$ 个节目在缓存中分配的前缀为$j_{i+1}-j_i$段。

4　性能评价

4.1　仿真环境与评价指标

前言介绍了几种已有的缓存分配算法，但这些算法都不适用于本文介绍的移动流媒体系统，因此无法与本文提出的缓存分配算法进行性能比较。为了评价本文的分配算法的性能，本文准备将其与文献[6]中用于进行性能比较的比例优先(Proportional Priority)分配算法[7]，简记为PP算法，进行比较。PP算法中，某个节目缓存的前缀大小与节目的大小与流行度的乘积成比例，同时前缀不会超过节目的大小。

为了评价算法的性能，我们设计了一个事件驱动的仿真系统，包含流媒体服务器，代理服务器和移动终端 3 个子系统，用以模拟整个移动流媒体系统。

设流媒体服务器包含 200 个CBR流媒体节目，每个节目的长度均为 120min，数据率均为 1.5Mbps，其访问概率服从参数 $\theta=0.27$ 的zipf分布，并将节目按访问概率从大到小的顺序从 0 到 199 进行编号。代理服务器的缓存大小以缓存比例f (缓存大小占流媒体服务器上流媒体总数据量的比值)的形式表示，缓存分段长度seg为 5M。仿真系统由移动终端用户的点播请求驱动运行，请求速率 λ 为 0.1/s，全部请求数为 5000 个。所有移动终端存储空间大小的最小值m为容纳90s的流媒体数据所需的空间大小，为 16.9M。阈值D_{th}为 30s，传输延时delay为 10s。p_c和p_d分别为 0.6 和 0.4，c_s为 1 单位，c_p为 0.2 单位。

在仿真系统启动时，代理服务器根据某种缓存分配算法将缓存空间分配给各个流媒体节目以保存前缀。系统运行过程中，当代理服务器接收到用户的点播请求后，首先查询是否有所请求节目的前缀在缓存中，如有，则直接发送前缀到移动终端播放，当前缀发送完毕或者缓存中没有前缀时，则从流媒体服务器向移动终端发送流媒体数据。

我们将比较各种算法下缓存的分配结果，同时采用单位时间(1s)内系统的总的平均传输成本、平均启动延时、综合指标(IIndex)和 BHR 等指标来评价系统在各种算法下的性能。

4.2　仿真结果

图 2 为在两种缓存大小(以缓存比例 f 表示)下，当 $\lambda=0.1$/s 时，移动流媒体系统采用本文提出的分配算法时与采用 PP 算法时缓存的分配结果的比较。随着 f 的增大，更多的空间分配给了更多更流行的节目。不同于 PP 算法只单纯考虑了流行度和节目大小，本文的算法的目标是使得综合了网络传输成本和启动延时的综合节省值达到最大，由于综合节省值与前缀长度和移动终端存储空间的大小都密切相关，因此缓存的分配结果与 PP 算法具有截然不同的特点，从而可以使总的综合节省值更大，系统性能更优。

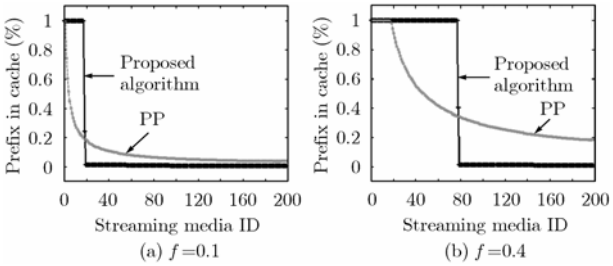

图 2 不同分配算法下的缓存分配结果

图 3 为在不同的缓存大小下，当系统采用本文提出的算法时与采用 PP 算法时，单位时间内所有节目的综合指标(IIndex)之和的比较。图 4 为在不同的缓存大小下，当系统采用本文提出的算法时与采用 PP 算法时，单位时间内所有节目的平均传输成本之和的比较。可以看出，与 PP 算法相比，本文的算法可以显著地减小综合指标和网络传输成本，当 f 为 10%时，综合指标之和比 PP 算法节省 14.4%，传输成本之和比 PP 算法节省 14.4%；当 f 为 50%时，综合指标之和比 PP 算法节省 17.4%，传输成本之和比 PP 算法节省 17.4%。这是因为，本文的算法明确地将最大化所有节目的综合节省值之和，最小化所有节目的综合指标之和作为目标，而且，传输成本是组成综合指标的重要部分，在最小化系统的综合指标之和的同时，也使得系统的传输成本之和减小。

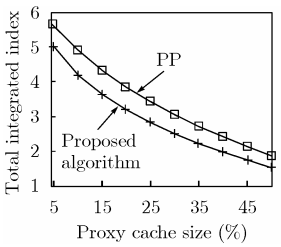

图 3 综合指标比较

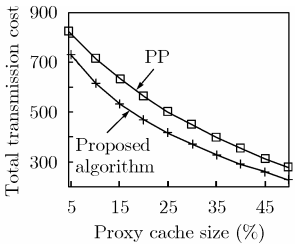

图 4 网络传输成本比较

图 5 为在不同的缓存大小下，当系统采用本文提出的算法时与采用PP算法时，单位时间内所有节目的启动延时之和的比较。在f大于 5%时，本文的算法和PP算法的启动延时都始终为 0。这是因为PP算法保证了为每个节目都在缓存中分配空间以保存前缀，当f大于 5%时，每个节目分配到的缓存所能容纳的前缀长度都大于阈值D_{th}，因此启动延时始终为 0；而本文的算法综合考虑了减小传输成本和启动延时，当p_c和p_d分别为 0.6 和 0.4，启动延时节省值被赋予了足够大的权值时，系统的启动延时也始终为 0，达到了最小。

图 6 为在不同的缓存大小下，当系统采用本文提出的算法时与采用 PP 算法时的平均 BHR 的比较。可以看出，与 PP 算法相比，本文的算法可以显著地增加平均 BHR，当 f 为 10%时，BHR 比 PP 算法增加 39.7%；当 f 为 50%时，BHR 比 PP 算法增加 6.3%。

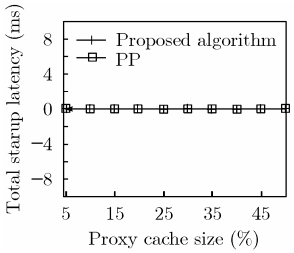

图 5 播放启动延时比较

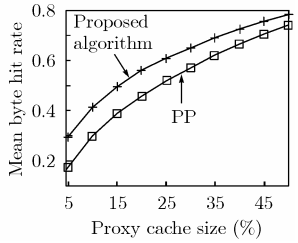

图 6 字节命中率比较

根据以上的仿真结果和分析，我们发现，本文提出的适用于移动流媒体系统的缓存分配算法比传统的PP算法在综合指标、网络传输成本和字节命中率等指标上都有显著的性能提高，这证明了本文的算法是非常有效的，不仅适用于移动流媒体系统，而且达到了我们的设计目的，有效地提高了系统的性能。另外，当p_c和p_d适当取值时，所有节目的播放启动延时为 0，达到了最小。这说明了本文的算法在选择适当参数时，也可以使得启动延时指标达到最优。

5 结束语

移动流媒体系统中代理服务器上的缓存分配算法是移动流媒体系统的关键算法，然而，到目前为止并没有学者提出适用于移动流媒体系统的缓存分配算法。本文提出了适用于移动流媒体系统的 MBatching 传输方案，推导出了在此方案下网络传输成本和移动终端播放启动延时的计算公式，并提出了基于动态规划法的适用于移动流媒体系统的分配算法。仿真结果表明，本文的算法完全适用于移动流媒体系统，可显著减少系统的网络传输成本和性能评价综合指标，可增加流媒体系统的字节命中率，可显著提高移动流媒体系统的性能，对移动流媒体系统的部署和研究具有重要的实际意义。

参 考 文 献

[1] Elsen I, Hartung F, and Horn U, *et al*.. Streaming technology in 3G mobile communication systems. *IEEE Computer Magazine*, 2001, 34(9): 46–52.

[2] Sen S, Rexford J, and Towsly D. Proxy prefix caching for multimedia streams. Proc. IEEE Infocom, New York, 1999: 1310–1319.

[3] Zhang Z, Wang Y, and DU D H C, *et al*.. Video staging: A proxy-server-based approach to end-to-end video delivery over wide-area networks. *IEEE/ACM Trans. on Networking*, 2000, 8(4): 429–442.

[4] Ramesh S, Rhee I, and Guo K. Multicast with cache(mcache): An adaptive zero-delay video-on-demand service, *IEEE Trans. on Circuits and Systems for Video Technology*, 2001, 11(3): 440–456.

[5] Kangasharju J, Hartanto F, and Reisslein M, *et al*.. Distributing layered encoded video through caches. *IEEE Trans. on Computers*, 2002, 51(6): 622–636.

[6] Wang B, Sen S, and Adler M, *et al*.. Optimal proxy cache allocation for efficient streaming media distribution. *IEEE Trans. on Multimedia*, 2004, 6(2): 366–374.

[7] Verscheure O, Venkatramani C, and Frossard P, *et al*.. Joint server scheduling and proxy caching for video delivery. *Computer Communications*, 2002, 25(4): 413–423.

雷正雄： 男，1978 年生，博士生，研究方向为下一代网络、移动流媒体技术.

廖建新： 男，1965 年生，教授，博士生导师，研究方向为移动智能网、下一代网络、流媒体技术.

朱晓民： 男，1974 年生，讲师，博士，研究方向为移动智能网、下一代网络.