

# 一种基于期望失真的 P2P 流媒体数据调度算法<sup>\*</sup>

刘亚杰 孙 晓 王 晖

(国防科技大学信息系统与管理学院 长沙 410073)

**摘 要** 数据调度问题是 P2P 流媒体研究中的核心问题。本文考虑 Peer 结点在带宽资源等方面的异构性,以分层编码为基础,提出了一种基于期望失真的数据包调度算法。它用期望失真来表示每个待调度数据包的重要程度,并在数据调度过程中优先请求调度那些期望失真值较大的数据包,从而可在接收节点上维持较高的播放质量水平。仿真实验表明,在多种实验参数条件下该调度算法均具有较好的性能。

**关键词** P2P 流媒体,异构,期望失真,调度算法

## An Expected-Distortion Based Data Scheduling Algorithm for Heterogeneous P2P Media Streaming

LIU Ya-Jie SUN Xiao WANG Hui

(College of Information System and Management, National Univ. of Defense Technology, Changsha 410073)

**Abstract** Data scheduling is a key problem in Peer-to-Peer media streaming research area. Considering the heterogeneities of peer nodes such as the peer nodes' bandwidth and so on, this paper proposes an expected-distortion based packet scheduling algorithm based on layered video coding. It uses expected-distortion to denote a packet's importance, and in the scheduling process it selects the most "important" packets to be requested at any given time, and thus to remain good playback quality on each receiver side. Simulation results show that under different experiment situations the proposed algorithm has good performance.

**Keywords** P2P media streaming, Heterogeneous, Expected-distortion, Scheduling algorithm

P2P 网络中的 Peer 结点在资源能力等方面具有明显的异构性,这种异构性体现在以下两个方面:1)从服务消费者角度来看,不同 Peer 结点的下行带宽和主机处理能力不同,导致它们应该接受和处理不同质量等级的流媒体数据;2)从服务提供者角度来看,不同 Peer 结点的上行带宽和缓存内容也具有差异性,即具有不同的服务能力,可提供不同数据内容的服务。而随着因特网和手机移动网的融合,大量移动终端会相继接入系统,上述异构性会更加明显。因此,迫切需要在 P2P 流媒体架构上根据节点的资源异构情况提供与之相匹配的服务,从而也提高系统的整体性能。

为适应 Peer 结点的异构性,有关研究人员提出了 MDC 编码与多组播树相结合的多种方案<sup>[1~3]</sup>,其主要原理为在服务器端把要发布的节目流分割成多条 MDC 子流,每条子流都用独立的组播树进行分发传输,Peer 结点可以根据自己的带宽资源情况选择加入到单棵或多棵组播树中;对任意结点而言,其接收并解码的 MDC 子流数越多,则播放质量越好。但多组播树的维护管理比较复杂,MDC 编码的效率也比较低。目前的研究主流<sup>[4~6]</sup>及其应用系统如 CoolStreaming<sup>[4]</sup>、GridMedia<sup>[5]</sup>等都建立在结点随机逻辑拓扑构建方法基础上,即每个新结点在加入系统时均通过某种集中或分布式索引机制从系统中随机获取一定数目的结点,和这些结点建立邻居关系,并采用相应的数据调度算法从邻居结点调度获取数据。但这类研究及其应用目前都还缺乏有效的结点异构性支持机制。

分层编码可适应 Peer 结点的异构性。与 MDC 编码不同

的是,分层编码中不同数据层在解码时存在依赖关系,且分层编码一般比 MDC 编码效率更高。本文基于分层编码,在以 Peer 接收结点为驱动、允许数据包适度重传的调度框架下,提出了一种数据包调度算法,它不仅充分考虑了 Peer 节点的异构性,并在满足不同传输路径上网络带宽大小等约束的同时,优化了 Peer 接收节点上的播放质量。本文研究不涉及 Peer 服务结点的搜索。

## 1 调度框架

先假设如下:对每个 Peer 接收结点而言,都存在多个 Peer 服务节点可为之提供数据服务,Peer 服务节点和 Peer 接收节点之间均存在单独的传输路径并具有不同的带宽、延迟、丢包率等特征,这些传输路径之间可能存在共享的子路径;无论是 Peer 服务节点还是 Peer 接收节点均不能预先知道相关网络传输路径上的带宽情况,Peer 节点可随时离开系统。

调度框架以 Peer 接收结点为驱动,该结点在整个调度过程中负责统计检测其与每个 Peer 服务节点之间网络路径上的丢包率、延迟、带宽等网络参数,并建立一滑动窗口,该窗口每隔时间向前滑动一次,其目的是确定下一时间间隔内从 Peer 接收节点允许请求调度的数据包;滑动窗口每滑动一次,则启动一次调度时机,并运行一次基于期望失真的数据包调度算法,以决定向哪些 Peer 服务节点发送哪些数据包调度请求。Peer 服务节点的主要任务是收到请求报文后随即发送相应的数据包报文,并定期向 Peer 接收结点发送其所缓存数据内容的信息。

<sup>\*</sup> Supported by the National Natural Science Foundation of China under Grant(No. 60433040)。刘亚杰 博士,讲师,主要研究方向为多媒体网络通信。

## 2 目标问题及调度算法

用  $M$  表示 Peer 服务节点的个数,  $N$  表示每个数据包被允许调度的时机数,  $F$  表示目标流所包含的帧数,  $L_i$  表示第  $i$  帧所包含的数据层数,  $l_{i,j}$  表示第  $i$  帧属于数据层  $j$  的数据包,  $s_{i,j}$  表示数据包  $l_{i,j}$  的大小,  $\Delta d_{i,j}$  表示  $l_{i,j}$  的解码对播放失真减少程度的度量,  $A_{i,j}$  表示  $l_{i,j}$  在解码过程中所依赖的数据包集合,  $B_{i,j}$  表示在解码过程中依赖于  $l_{i,j}$  的数据包集合,  $t_{DTS,l_{i,j}}$  表示  $l_{i,j}$  在接收结点的解码期限,  $t_{0,l_{i,j}}, \dots, t_{N-1,l_{i,j}}$  表示  $l_{i,j}$  的  $N$  次连续调度时机,  $a_{i,j}$  表示  $l_{i,j}$  是否在接收结点被解码,  $t_{cur}$  表示当前调度时机,  $G_{cur}$  表示当前调度时机 Peer 接收节点允许调度的候选数据包集合, 那么数据包的调度优化目标可表示为

$$\text{Min} \sum_{i=1}^F \sum_{j=1}^{L_i} \Delta d_{i,j} - \sum_{i=1}^F \sum_{j=1}^{L_i} a_{i,j} \Delta d_{i,j} \quad (1)$$

并满足 Peer 接收节点到各个 Peer 服务节点之间可用带宽的约束以及各个 Peer 服务节点所缓存数据内容的约束。式(1)代表最小化接收节点的播放失真, 即最大化其播放质量。

式(1)面向的是目标节目的全部数据。但在每个调度时机, 由于 Peer 服务节点的缓存内容范围有限, 因此对数据包的调度范围也只能局限于部分节目数据, 由此可把式(1)所代表的目标近似转化为在每个调度时机对目标节目局部数据进行调度, 以优化 Peer 接收节点的播放质量。在每个调度时机, 考虑分层编码数据包之间所存在的解码依赖关系、网络带宽资源限制、数据传输过程中的丢包、延时等多种因素, 如何对数据包进行优化调度也难以进行直观决策。由于分层编码中不同数据包对播放质量的贡献不同, 这里提出一种启发式调度算法, 它通过计算  $G_{cur}$  中每个数据包的期望失真, 并在满足相关约束的前提下, 优先向 Peer 服务节点请求调度那些期望失真值较大的数据包, 从而提高 Peer 接收节点的播放质量。下面介绍如何在多 Peer 服务节点情形下计算每个数据包的期望失真。

对  $l_{i,j} \in G_{cur}$ , 从解码依赖关系角度考虑如下三种情形: 1) 如果集合  $A_{i,j}$  中有成员还未被调度请求, 由于  $l_{i,j}$  的解码依赖  $A_{i,j}$  中的所有数据包, 则  $l_{i,j}$  在当前调度时机被请求的紧迫性或重要性应很小, 即其期望失真值也应该较小; 2) 如果  $A_{i,j}$  中所有成员已被调度请求, 则  $l_{i,j}$  被调度的重要性应随  $A_{i,j}$  中所有成员能按期到达 Peer 接收节点的概率的增大而增加, 即其期望失真也应随之增大; 3) 如果  $B_{i,j}$  中有成员已被调度请求, 由于这些成员的解码依赖于  $l_{i,j}$ , 那么  $l_{i,j}$  被请求的重要性应进一步增加, 即其期望失真也应进一步增大。

对集合  $A_{i,j} \cup B_{i,j}$  的任意数据包成员  $l_{r,j}$ , 用  $\epsilon_{r,j}^H$  表示不能在其解码期限之前到达 Peer 接收节点的概率, 并约定该概率的计算仅基于发生在  $l_{r,j}$  上的历史调度行为以及  $l_{r,j}$  在 Peer 接收节点上的当前状态, 即如果  $l_{r,j}$  的当前状态为未请求, 则  $\epsilon_{r,j}^H$  等于 1; 如果为已接收, 则  $\epsilon_{r,j}^H$  等于 0; 如果为已请求但未接收, 则进一步分两种情况讨论:

(i) 如果  $t_{DTS,l_{r,j}} \leq t_{cur}$ , 即  $l_{r,j}$  未能在其解码期限前到达 Peer 接收节点, 则  $\epsilon_{r,j}^H$  等于 1;

(ii) 如果  $t_{DTS,l_{r,j}} > t_{cur}$ , 设  $l_{r,j}$  的历史调度时机序列为  $t_{0,l_{r,j}}, \dots, t_{j-1,l_{r,j}}$  ( $j \leq N$ ), 用 0-1 变量  $a_{k,l_{r,j}}^m$  表示是否在调度时机  $t_{k,l_{r,j}}$  ( $0 \leq k \leq j-1$ ) 向 Peer 服务节点  $m$  ( $m=0, \dots, M-1$ ) 发送了对数据包  $l_{r,j}$  的请求, 那么该情况下  $\epsilon_{r,j}^H$  可表示为

$$\epsilon_{r,j}^H = \prod_{k=0, m, a_{k,l_{r,j}}^m = 1}^{j-1} \prod P\{RTT^m > t_{DTS,l_{r,j}} - t_{k,l_{r,j}}\} \quad (2)$$

对数据包  $l_{i,j}$  的期望失真  $\Delta \tilde{d}_{i,j}$ , 考虑上述三种情形及  $\epsilon_{r,j}^H$  的表达式, 可定义  $\Delta \tilde{d}_{i,j}$  的表达式如下:

$$\Delta \tilde{d}_{i,j} = ((1 - \epsilon_{i,j}^H) \prod_{l_{r,j} \in A_{i,j}} (1 - \epsilon_{r,j}^H)) (\Delta d_{i,j} + \sum_{l_{r,j} \in B_{i,j}} \Delta d_{r,j} (1 - \epsilon_{r,j}^H)) \quad (3)$$

其中,  $\prod_{l_{r,j} \in A_{i,j}} (1 - \epsilon_{r,j}^H)$  为集合  $A_{i,j}$  中所有成员能按期到达 Peer 接收节点的概率,  $(1 - \epsilon_{i,j}^H)$  为  $l_{i,j}$  能按时到达 Peer 接收节点的概率, 把二者的乘积作为式(4~12)中的乘法因子, 则反映了上述第 1)、2) 种情形。而  $\Delta d_{i,j}$  与  $\sum_{l_{r,j} \in B_{i,j}} \Delta d_{r,j} (1 - \epsilon_{r,j}^H)$  的相加, 不仅反映了  $l_{i,j}$  本身的失真值, 还反映上述第 3) 种情形。

再考虑剩余调度时机中  $l_{i,j}$  上可能发生的请求调度行为, 即如果  $l_{i,j}$  上剩余的调度时机数越小, 其在当前调度时机被请求的概率应越大, 即其期望失真也应该越大。设  $l_{i,j}$  上剩余的调度时机序列为  $t_{j+1,l_{i,j}}, \dots, t_{N-1,l_{i,j}}$ , 用  $\epsilon_{i,j}^X$  表示仅基于这些调度时机组合所导致的不能使  $l_{i,j}$  按期到达 Peer 接收节点的最小概率, 用  $P_{i,j}$  表示当前缓存了数据包  $l_{i,j}$  的 Peer 服务节点集合, 则  $\epsilon_{i,j}^X$  可表示为

$$\epsilon_{i,j}^X = \prod_{k=j+1}^{N-1} \left( \sum_{m: l_{i,j} \in P_{i,j}} P\{RTT^m > t_{DTS,l_{i,j}} - t_{k,l_{i,j}}\} \right) \quad (4)$$

最后结合式(3)和(4), 可重新调整  $\Delta \tilde{d}_{i,j}$  的定义如下:

$$\Delta \tilde{d}_{i,j} = ((1 - \epsilon_{i,j}^H) \prod_{l_{r,j} \in A_{i,j}} (1 - \epsilon_{r,j}^H)) (\Delta d_{i,j} + \sum_{l_{r,j} \in B_{i,j}} \Delta d_{r,j} (1 - \epsilon_{r,j}^H)) \epsilon_{i,j}^X \quad (5)$$

## 3 仿真试验及性能分析

我们使用 ns-2 模拟器对其进行了仿真实验评估。实验中所采用的网络拓扑结构及相关的参数如图 1 所示, 其中  $P_1$ 、 $P_2$  和  $P_3$  分别为三个不同的 Peer 服务节点,  $R$  为 Peer 接收节点, 在连接  $P_i$  ( $i=1, 2, 3$ ) 和  $R$  的网络路径上均包含了路由器  $R_1$  和  $R_2$ , 并在  $R_1$  和  $R$  之间的路由路径上形成共享链路。图 1 也标出了每条网络链路上的传输延迟、带宽大小等属性。

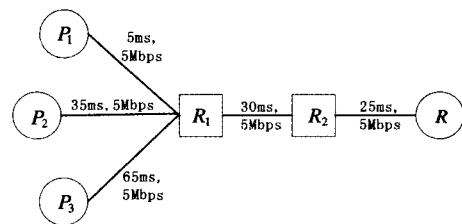


图 1 仿真试验的拓扑及参数

实验中所使用的视频流是 stefan, 其格式为 CIF (352288), 共 600 帧, 并用 MPEG-4 FGS 编码/解码工具<sup>[7]</sup>对它进行了压缩, 压缩后的播放帧率为 10 帧/秒, 数据包的大小设置为 1000 字节; 在实验开始之前, 先通过 MPEG-4 FGS 解码器计算获取了每个数据包的失真值: 对任意帧的基础层, 如果其大小为多个数据包, 则每个数据包的失真值为该基础层总的失真值与其对应的数据包个数的比例; 对于任意帧的增强层, 则按被打包的个数把它划分为多个增强子层。我们采用峰值信噪比 (PSNR) 来作为 Peer 接收节点播放质量的评价标准, 并把本文所提的基于期望失真的调度算法与 PALS<sup>[8]</sup>中所提调度算法进行比较。PALS 采用“之”字形调度策略,

以体现不同层次的数据包在解码时存在的依赖关系,总体属于一种定性的分析调度;而本文对每一个数据包的重要程度进行了量化。

首先不考虑 Peer 节点的离开行为,且假设所有 Peer 服务节点上都缓存了目标流中所有数据,在不同的播放启动延迟条件下对两种算法的性能进行了比较,其结果如图 2 所示,这里  $\Delta$  设置为 500ms。从图 2 可以看出,播放启动延迟越大,二者所获得的平均播放质量越好;在不同的启动延迟条件下,对比 PALS 中所提出的调度算法,采用本文所提的算法均能够使 Peer 接收节点的播放质量得到一定程度的提高;此外,播放启动延迟越大,两种调度算法所获得的平均播放质量的差距也越小。

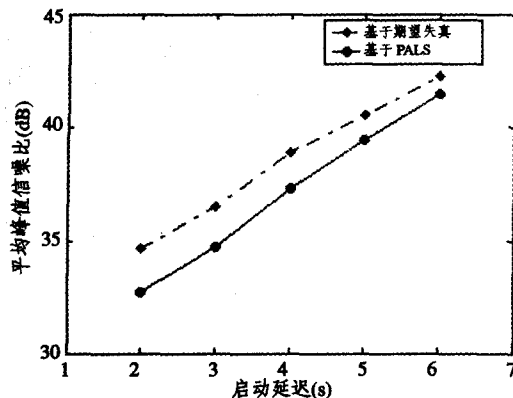


图 2 不同播放延迟下 Peer 接收节点播放质量比较

为了考察 Peer 服务节点的离开行为对 Peer 接收节点播放质量的影响,在  $\Delta=500\text{ms}$  及播放启动延迟等于 4s 的设置前提下,分别运行两种算法,且当仿真实验运行到 32s 时我们让 Peer 服务节点  $P_3$  离开,并记录 Peer 接收节点上所接收到的每个帧的 PSNR 值。图 3 给出了记录结果,可以看出,当节点  $P_3$  离开后,两种算法情况下 Peer 接收节点所获得的播放质量会有小幅下降。通过观察 Peer 接收节点实际获得的有效下载带宽,我们发现当  $P_3$  离开后,Peer 接收节点的有效下载带宽减少,从而使得其播放质量出现一定程度的下降。从图 3 也可以看出,在节点  $P_3$  离开前后,对比 PALS 中所提算法,本文所提算法均能够使 Peer 接收节点的播放质量得到一定程度的提高。

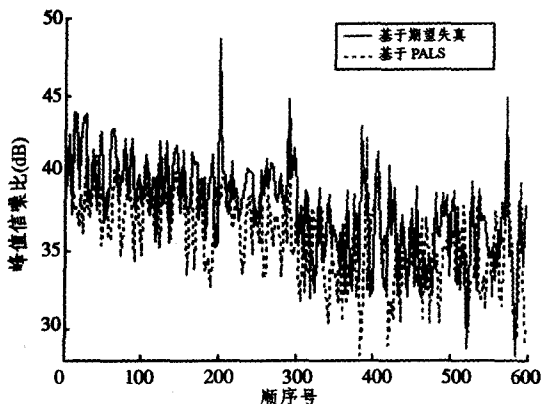


图 3 有节点离开情况下播放质量的比较

最后,基于同样的假设,但认为 Peer 服务节点所缓存数据异构,从而对 Peer 接收节点所获得的播放质量进行评估。

这里假设节点  $P_1$  上缓存了所有层次的数据,节点  $P_2$  仅缓存了增强层中从低层到高层顺序 2/3 数量的数据子层,节点  $P_3$  上仅缓存了增强层中从低层到高层顺序 1/3 数量的数据子层,分别运行两种算法,并在实验中记录 Peer 接收节点上所接收到的每个帧的 PSNR 值。图 4 给出了记录结果,从图 4 可以看出,当 Peer 服务节点所缓存的数据异构时,对比 PALS 中所提算法,本文所提算法也能够使 Peer 接收节点的播放质量得到一定程度的提高。

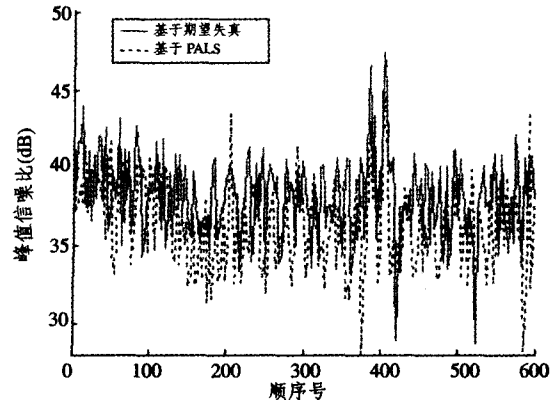


图 4 节点缓存内容异构情况下播放质量的比较

结论 本文考虑 Peer 节点的异构性,在分层编码的基础上提出了一种基于期望失真的数据包调度算法。在对期望失真的表达式进行推导计算的过程中则综合考虑了数据包解码依赖关系、数据包对解码失真的贡献、数据包解码期限要求以及不同网络传输路径上的丢包、延迟等多种因素。仿真实验表明,对比现有的类似调度算法,本文所提出的基于期望失真的调度算法在多种仿真实验条件下均能够提高 Peer 接收节点的播放质量。

## 参考文献

- 1 Padmanabhan V N, Wang H J, Chou P A. Resilient Peer-to-Peer Streaming. In: IEEE International Conference on Network Protocols. Atlanta, GA, USA, 2003
- 2 Castro M, et al. SplitStream: High-bandwidth multicast in a cooperative environment. In: 19th ACM Symposium on Operating Systems Principles (SOSP), 2003
- 3 Venkataraman V, Yoshida K, Francis P. Chunkyspread: Heterogeneous Unstructured Tree-Based Peer-to-Peer Multicast. ICNP Santa Barbara, California, USA, 2006
- 4 Zhang X, et al. CoolStreaming/DONet: A Data-driven Overlay Network for Live Media Streaming. In: IEEE INFOCOM, Miami, FL, USA, 2005
- 5 Zhang M, et al. Large-Scale Live Media Streaming over Peer-to-Peer Networks through Global Internet. P2PMMS'05, Singapore, 2005
- 6 Zhang M, et al. On the optimal scheduling for media streaming in data-driven overlay networks. In: IEEE Globecom, San Francisco, California, USA, 2006
- 7 Radha H M, Schaar M v d, Chen Y. The MPEG-4 Fine Grained Scalable Video Coding Method for Multimedia Streaming Over IP. IEEE Transaction on Multimedia, 2001, 3(1): 53~68
- 8 Rejaie R, Ortega A. PALS: Peer-to-Peer Adaptive Layered Streaming. In: International Workshop on Network and Operating Systems Support for Digital Audio and Video, Monterey, California, 2003