

ARLNCStream:自适应随机网络编码流媒体系统

武广柱 王劲林 齐卫宁

(中国科学院声学研究所 DSP 中心 北京 100080)

摘要: 该文设计了一种自适应随机线性网络编码 P2P 实时媒体分发系统。提出了一种网状网络拓扑下的编码窗自适应随机线性网络编码算法。仿真结果表明, 该算法具有很好的效果: 对网络带宽要求低; 无需全局网络拓扑信息; 平均延迟小, 连续性好; 鲁棒, 可扩展性强。

关键词: 自适应随机线性网络编码; P2P; 媒体分发; 网络编码

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2008)01-0025-04

ARLNCStream: Adaptive Random Linear Network Coding for Media Streaming System

Wu Guang-zhu Wang Jin-lin Qi Wei-ning

(Institute of Acoustics, Chinese Academy of Sciences, Beijing 100080, China)

Abstract: In this paper, ARLNCStream, a novel algorithm of adaptive random linear network coding for media streaming, is presented. When distributing contents using random linear network coding, every node adjusts its coding window in an adaptive way and then requests coded block of an appropriate size from its partners. The solution makes nodes collaborate well with potential partners. Without knowledge of global network topology, ARLNCStream archives quiet low segment missing ratio. ARLNCStream is also robust and scalable. Simulations demonstrate the excellent performance of the solution.

Key words: Adaptive random linear network coding; Peer-to-Peer; Media distribution; Network coding

1 引言

支持大规模高并发特性的实时媒体分发系统是当前研究的一个热点。由于节点动态、网络异构、单点瓶颈等 P2P 网络特性, 加上用户对媒体的实时性要求, 虽然先前研究^[1-9]已经取得了许多成果, 但仍未能解决很多本质问题, 如延迟长、控制开销大、系统对网络要求高甚至单点失效等。

本文受 CoolStreaming^[1]启发, 并根据随机线性网络编码理论, 结合实时媒体分发系统的特点, 提出了一种网状网络拓扑结构下的自适应随机网络编码实时媒体分发算法: ARLNCStream(Adaptive Random Linear Network Coding for Media Streaming)。仿真结果表明, ARLNCStream 具有对网络状况要求低、无需全局网络拓扑信息、平均延迟小连续性好、鲁棒性高可扩展性强的特点。

本文组织如下: 第 2 节概述了实时媒体分发系统的相关工作; 第 3 节介绍了 ARLNCStream 的算法细节; 第 4 节给出了仿真结果及其分析; 第 5 节是结束语。

2 相关工作

按网络拓扑结构划分, 实时媒体分发系统大体可以分为 3 种类型: 单分发树模型、多分发树模型以及网状结构模型。

NICE^[2], ZIGZAG^[3]等系统采用单转发树协议。这类系统在覆盖层建立单转发树, 并通过动态聚类形成一些节点簇。算法限制树的度数, 并采用一系列措施来维持转发树的完整性。实验表明^[2,3], 此类系统具有较好的效果。虽然这类系统都具有抗毁算法, 但对单点失效仍然敏感。父节点的失效会给下游节点带来不同程度的影响, 甚至会严重影响媒体播放的流畅性。

许多系统如 Bullet^[4], SplitStream^[5], PALS^[6], CoopNet^[7]等通过建立多分发树或利用多描述编码算法来提高系统的鲁棒性及节点负载的均衡性, 但仍未解决垂直连接带来的路径瓶颈问题。

CoolStreaming/DONe^[1]提出了一种基于 Gossip 机制的数据驱动分发系统。这类系统不再专门设计用于构造和维持分发树的算法, 而是采了网状结构模型。节点随机选择合作节点, 构成网状协作结构。这种结构通过使用更多的连接带来了并行下载的优点, 提高了服务质量。此类系统往往采用复杂的数据块调度算法, 目的是将数据块均匀分布在网络中, 提高请求命中率, 以达到更好的并行效果。BiTos^[8]将 BitTorrent^[9]的算法做了很小的调整, 用于实时流媒体分发系统。其思想是将需要下载的媒体文件片段分成紧急片段和一般片段, 通过一个随机因子来控制下载哪种片段, 以此来达到播放的连续性要求并获得了 BitTorrent^[9]算法带来的优点。

2006-06-23 收到, 2006-11-15 改回

中国下一代互联网示范工程(CNGI)视频多媒体点播系统(CNGI-04-15-2A)资助课题

本文对异构对等网络环境下的实时媒体分发算法进行了研究,利用网络编码过程引入的分布随机性避免了复杂的数据块调度算法,取得了很好的效果。

3 ARLNCStream

香农在 1956 年就指出:通信网络端对端的最大信息流,是由网络有向图的最小割决定的。在传统网络的信息传输过程中,中间节点除存储转发外,一般不做其他形式的数据处理。人们普遍认为,中间节点所进行的数据处理并不会给数据传输过程本身带来任何好处。存储转发模式使这一最大流最小割定理的上限在很长时间内显得遥不可及。

Ahlsvede 等人提出的网络编码^[10]指出:允许中间节点对信息流进行编码组合可以达到最大流最小割定理的上界。并在此基础上进一步证明,在单源多宿情况下,采用线性网络编码一定能够达到该上界。网络编码代表了一种协同工作的理念,这使得它的应用不再局限于改进多播和增加网络容量。网络编码已与其它技术相结合应用于网络管理、纠错、信息安全、对等网络通信、路由和交换等数十个领域。

在网络编码应用于对等网络方面,文献[11]将网络编码用于 P2P 文件分发。与传统非编码方式相比,其算法取得了节省服务器带宽 20~30%和提高下载速度 2~3 倍的性能。但是,由于需要将整个文件下载完毕后才能进行解码,文中所提的算法并不适合实时媒体分发系统。

受 CoolStreaming^[1]算法的启发,本文将文件按固定大小进行片段划分,由若干个片段组成文件块,并对每个块按片进行随机网络编码。节点随机选择合适的合作节点,构成网状拓扑结构,并从合作节点下载经随机网络编码后的文件块。当一个节点收集到足够多的信息可以对某块进行解码时,节点解并将解码结果放到播放缓冲区。

如图 1 所示,节点 A 的缓存中含有已解码的片段 2,3 和未解码的片段 4,5 的编码信息;节点 B 有已解码的片段 5,6 和未解码的片段 7,8 的编码信息;节点 C 有已解码的片段 3 和未解码的片段 4,5,6 的编码信息。节点 D 已经有已解码的片段 2,3,现在希望下载片段 4,5,6。假设节点 D 向节点 A 发出请求,节点 A 收到请求后检查自身缓存,发现缓存中有片段 4,5 的编码信息 $\text{Code}(p_4, p_5) = (\eta_1, \eta_2) * (p_4, p_5)^T$ 而缺少片段 6,于是节点 A 将 $\text{Code}(p_4, p_5)$ 乘以一个非零随机数 rand_1 并扩充编码为 $\text{Code}(p_4, p_5, p_6) = \text{rand}_1 * (\eta_1, \eta_2, \mathbf{0}) * (p_4, p_5, p_6)^T$ 后发送给节点 D。同样,上述中 p_{-1} 代表本片段缺失, p_i 代表片段 i 。同样, B 发送 $\text{Code}(p_4, p_5, p_6) = (\mathbf{0}, \alpha_1, \alpha_2) * (p_{-1}, p_5, p_6)^T$ 给 D, C 发送 $\text{Code}(p_4, p_5, p_6) = \text{rand}_2 * (\beta_1, \beta_2, \beta_3) * (p_4, p_5, p_6)^T$ 给 D。只要三个编码向量 $(\mathbf{0}, \alpha_1, \alpha_2)$, $(\beta_1, \beta_2, \beta_3)$, $(\eta_1, \eta_2, \mathbf{0})$ 线性无关,则节点 D 可以通过矩阵求逆来解码得到片段 4,5,6。

这种分块编码的方式,一方面节点可以较快解码而达到播放连续性的要求;另一方面利用网络编码使各片段较均匀

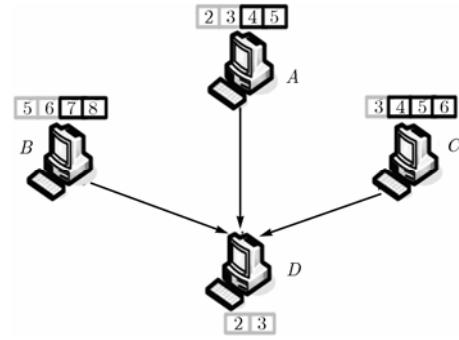


图 1

的随机分布在各个节点上,从而提高片段请求的命中率,同时也避免了通常的“最少最优”片段分配算法的局部优化而非全局最优的缺点。

为了验证这一方法的有效性,我们刻意选择较差的网络环境进行仿真。以 GT-ITM(Georgia Tech Internet Network Topology Models)生成网络拓扑,并随机选取了 100 个节点作为仿真对象。节点带宽在 2Mb~5Mb 间随机分布,服务器带宽为 10Mb。仿真中,媒体文件的大小为 300MB,码率 500kbps,分片大小为 256kB,块大小 12 片,节点播放缓冲区大小为 20s。每隔 1s 就有一个新节点向服务器发出加入请求,服务器收到请求后,给出最多为 30 个节点的候选节点列表。新加入节点从列表中随机选取部分节点,发出块请求报文。收到请求的节点检查其缓存后发送相应编码信息或者拒绝报文。当连续多次请求某块遭失败时,节点认为该块在网络中分布稀少,直接向服务器发出块请求。节点随时解码,当播放缓冲区填满后节点开始播放。仿真结果表明,各节点平均播放连续度可达 0.75(见图 2)。

仿真证明,分块随机网络编码可以用于实时媒体分发。在进一步的研究中发现,块大小的选择是影响系统性能的重要因素:块过小,不能充分发挥网络编码带来的效能;块过大,因节点需要较长时间来收集足够多的信息才能解码,造成播放不够流畅。块大小的选择需要考虑整个网络的拓扑结构、节点数、节点缓存和带宽等情况。由于缺乏全局信息,加上 P2P 网络的异构特性,固定一个合适的块大小是非常困难的:我们难以选择适合所有节点带宽、充分发挥网络编码效能并以较高概率保证播放连续性的块大小。

在 P2P 网络中,端点的带宽存在较大差异,各个端点的缓存内容和当前播放时间点也不同步。有些节点带宽较大,并且其缓存中具有足够多的可供继续播放的片段;但有些节点带宽较小,其缓存中也可能还没有即将播放的片段。对于那些带宽较高并具有足够多可供播放片段的节点,应该选择较大的编码窗;对于那些带宽较低并且缺少可供播放片段的节点,应该选择较小的编码窗。然而,窗的大小还不能由这两个条件完全确定,合适的窗的大小还受全局网络拓扑情况和节点数等的影响。

ARLNCStream 依据节点的平均下载速度和其缓存情况, 动态调整编码窗的大小。获取这两个条件无需全局信息, 但这两个条件却隐含了全局网络拓扑信息并反映出各节点的配合情况。这样, 依据这两个条件而寻找到一个好的自适应算法, 既可以避开获取全局信息的难题又能使系统中的节点达到较好的协作状态。下面给出 ARLNCStream 的工作过程及其采用的自适应算法。

节点首先决定编码窗口 W 的大小, 然后从候选节点中随机选择一个节点发出请求, 在收到编码信息后计算从发出请求到收到信息所用的真实下载时间 T_{real} , 然后随机延迟时间 T_{relay} 后再次发出请求。等待 T_{relay} 的目的是为减轻整个网络的压力, 因为媒体分发系统并非以最快分发给目的。

(1) 填充播放缓冲区并计算下载速度。节点加入系统后从候选节点集中随机选择一个节点向其发出请求, 编码窗 W 初值选择一个固定大小(本文选择为 8), 下载完毕并可解码一个块后, 计算下载速度:

$$W = P_{\text{init}} \quad (1)$$

$$S = P_{\text{init}}/T \quad (2)$$

(2) 调节编码窗, 准备下载下一个区块。如果缓冲区已经填充完毕, 则启动播放线程。

$$W_{\text{old}} = W \quad (3)$$

$$W = \beta\{[P_{\text{cachemax}}M/S_{\text{play}} - T_{\text{min}}]S\} + (1 - \beta)W_{\text{old}} \quad (4)$$

其中 β 是一个加权因子, 目的是限制 W 的剧烈变化, 研究中我们发现取值 0.5 左右较为合适。 P_{play} 为当前播放片段号, M 是每个片段的大小, S_{play} 为码率。 T_{min} 是播放缓冲内的最小预留时间, 选取这一冗余量是为了尽量保证整个播放期间缓冲内至少有可供播放 T_{min} 时间的内容, 从而在发生下载困难等情况时为处理赢得时间, 而不因缓存中没有可供播放的内容而迫使播放线程中断播放。在启动播放线程前, T_{min} 取值为 0, 直到 20s 的缓冲填充完毕。其余情况 T_{min} 取值为播放缓冲的四分之一左右(5s)。

(3) 向合作节点发送块请求, 下载编码向量。每成功获取一编码信息向量, 节点都要延迟一时间 T_{relay} 后再次发出请求, 直到块下载完成。 T_{relay} 在每次下载完毕一编码信息向量后都需要做动态调整。

$$T_{\text{relay}} = (W/S - T_{\text{elapsed}})/N - T_{\text{real}} \quad (5)$$

其中 T_{elapsed} 为下载本块已经耗费的时间, N 为解码当前块还需要的最少编码信息向量数, T_{real} 为下载本编码向量的实际耗费时间(不包含 T_{relay})。

(4) 下载完成块, 更新下载速度。

$$S_{\text{old}} = S \quad (6)$$

$$S_{\text{curr}} = W/T \quad (7)$$

$$S = \alpha S_{\text{old}} + (1 - \alpha)S_{\text{curr}} \quad (8)$$

其中 α 为加权因子, 目的是保持估算的下载速度 S 的相对稳定, 取值 0.5 左右较为合适。

(5) 转步骤 2, 直到下载整个媒体文件完毕或退出播放。

窗口大小过分分散, 会给并行下载带来负面影响, 为此, 本文采取了限制窗口大小过分分散的措施。窗口大小估算出来后要取距离 W 最近的且为 2 的倍数的整数作为窗口大小, 称为自适应模 2 窗。

每开始播放一个片段, 端点都要检查其缓存中的内容。如果下一个即将播放片段还不具备, 则估计当前的网络编码下载过程是否可以在播放到该片段时提供解码, 如果不能, 则随机选择一个端点请求这一即将播放的片段, 多次尝试仍未得到该片段后, 直接向服务器发紧急请求。

4 仿真结果

本文以 GT-ITM 生成一个具有 500 个节点的网络拓扑。设定服务器带宽为 10Mb, 节点带宽在一定范围内随机分布。从 500 个节点中随机选出的 100 个节点每隔 1s 加入分发系统。媒体文件大小为 300MB, 片段大小 256kB, 码率 500kbps, 节点缓存大小 20s。

为了与其他算法比较性能, 我们对基于 Gossip 的数据驱动算法^[1]进行了仿真。

图 2 为节点带宽在 2~5Mb 和 3~7Mb 随机分布时, 各节点在 2000s 中的平均播放连续性情况。虽然 2~5Mb 的网络条件很差, 自适应编码窗算法可使连续性在 0.8 左右。节点带宽在 3Mb 到 7Mb 随机分布时, 基于 Gossip 的数据驱动方式连续度在 0.82 左右, 而 3 种编码窗选择方法均可使播放连续性在 0.87 以上, 特别是采用了模 2 编码窗后, 播放连续性几乎保持在 1.0 水平上, 取得了非常好的效果。

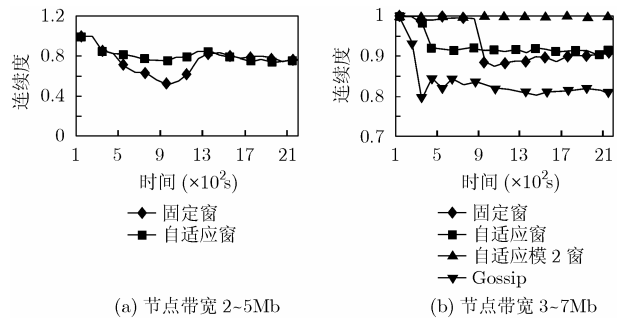


图 2 连续性

图 3 为节点带宽 3~7Mb 时的服务器负载情况, 自适应模 2 窗大大减轻了服务器的负载压力。ARLNCStream 具有很好的可扩展性。

图 4 是节点带宽 3~7Mb, 采用模 2 自适应窗编码时, 在 2000s 内的窗口大小分布情况。窗口大多分布在 6 到 14 之间, 没有出现大于 38 的窗口。正是由于各个节点动态自适应调整其编码窗大小, 优化了节点间的协作状态, 适应了特定的网络拓扑环境, 才使播放连续性达到 1.0。

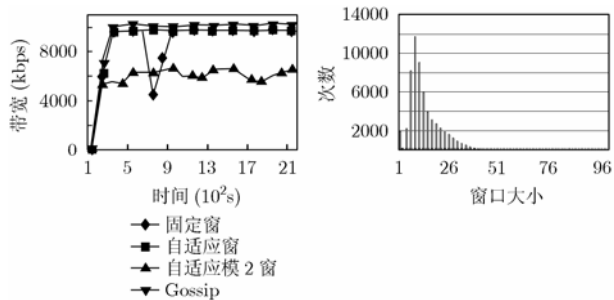


图 3 服务器带宽消耗

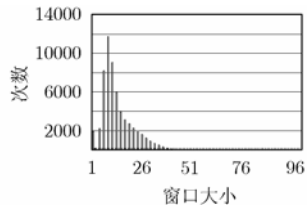


图 4 窗口大小分布

5 结束语

本文提出了一种全新的自适应随机线性网络编码实时媒体分发算法, 利用网络编码来均衡数据分布, 避免了复杂的数据块分配算法的同时提高了请求命中率; 利用自适应模 2 窗进行网络编码, 提高了播放连续性并大大降低了服务器带宽消耗。仿真结果也表明, 该算法在较差的网络环境下也能取得非常好的效果, 系统对网络状况要求较低。

在以后工作中, 我们将进一步优化合作节点的选取, 使得本自适应算法达到更加满意的效果。比如参数 α , β 选取问题, 我们研究中发现 0.5 左右效果较好, 但有待寻找其最优化的具体表达式。另外, 非法恶意节点加入系统后, 通过分发虚假编码信息, 可以致使系统中的节点解码错误, 从而无法播放。所以, 安全性问题也是需要我们进一步研究。

参考文献

- [1] Zhang Xinyan, Liu Jiangchuan, Li Bo, and Tak-Shing Peter Yum. CoolStreaming/DONet: A data-driven overlay network for efficient live media streaming. In Proceedings of IEEE INFOCOM, Miami, USA, March 2005, 3: 2102-2111.
- [2] Chu Y H, Rao S G, and Zhang H. A case for end system multicast. in Proc. SIGMETRICS'00, Santa Clara, California, United States, June 2000: 1-12.

- [3] Tran D A, Hua K A, and Do T T. A peer-to-peer architecture for media streaming. *IEEE J. on Select. Areas in Comm.*, 2004, 22(1): 121-133.
- [4] Kostic D, Rodriguez A, Albrecht J, and Vahdat A. Bullet: high bandwidth data dissemination using an overlay mesh. in Proc. ACM SOSP'03, New York, USA, October 2003: 282-297.
- [5] Castro M, Druschel P, Kermarrec A M, Nandi A, Rowstron A, and Singh A. SplitStream: high-bandwidth multicast in cooperative environment. in Proc. ACM SOSP'03, New York, USA, October 2003: 292-303.
- [6] Rejaie R and Ortega A. PLAS: peer to peer adaptive layered streaming. in Proc. NOSSDAV'03, Monterey, CA, USA, Jun. 2003: 153-161.
- [7] Padmanabhan V N, Wang H J, Chou P A, and Sripanidkulchai K. Distributing streaming media content using cooperative networking. in Proc. NOSSDAV'02, USA, May 2003: 177-186.
- [8] Vlavianos A, Iliofotou M, and Faloutsos M. BiToS: Enhancing bittorrent for supporting streaming applications. in 9th IEEE Global Internet Symposium 2006, Barcelona, Spain, April 2006: 33-39.
- [9] Cohen B. Incentives build robustness in bittorrent. in P2P Economics Workshop, Berkeley, CA, 2003: 5-6.
- [10] Ahlswede R, N Cai, Li S Y R, and Yeung R W. Network Information Flow. *IEEE Trans. on Information Theory*, 2000, 46(4): 1204-1216.
- [11] Gkantsidis C and Rodriguez P. Network coding for large scale content distribution. IEEE INFOCOM 2005, Miami, March 2005: 2235-2245.

武广柱: 男, 1979 年生, 博士生, 研究方向为宽带通信及嵌入式终端设备。

王劲林: 男, 1964 年生, 主任研究员, 博士生导师, 研究方向为宽带多媒体通信。

齐卫宁: 女, 1980 年生, 博士生, 研究方向为宽带通信。