

基于网络编码的对等网流媒体传输模型和算法*

张志明^{1,2+}, 周晋², 陈震^{2,3}, 李军^{2,3}

¹(清华大学 自动化系, 北京 100084)

²(清华大学 信息技术研究院, 北京 100084)

³(清华信息科学与技术国家实验室, 北京 100084)

Modeling and Algorithm for Network Coding Based P2P Streaming

ZHANG Zhi-Ming^{1,2+}, ZHOU Jin², CHEN Zhen^{2,3}, LI Jun^{2,3}

¹(Department of Automation, Tsinghua University, Beijing 100084, China)

²(Research Institute of Information Technology, Tsinghua University, Beijing 100084, China)

³(Tsinghua National Laboratory for Information Science and Technology, Beijing 100084, China)

+ Corresponding author: E-mail: zzm02@mails.tsinghua.edu.cn, <http://security.riit.tsinghua.edu.cn/wiki/NSLab>

Zhang ZM, Zhou J, Chen Z, Li J. Modeling and algorithm for network coding based P2P streaming. *Journal of Software*, 2012, 23(3): 648-661. <http://www.jos.org.cn/1000-9825/3991.htm>

Abstract: In P2P (peer-to-peer) streaming systems, server bandwidth consumption can be reduced by enhancing utilization ratio of nodes' (users') output (uplink) bandwidth capacity. With the ability of achieving maximum throughput of multicast, network coding has the potential to contribute to the enhancement. This article applies random linear network coding (RLNC) to P2P streaming system, and modeled transmission of P2P streaming. Greedy, rarest-first and random streaming algorithms are studied comparatively through streaming algorithm optimizations based on the framework of transmission model. Optimization results indicate that the random streaming algorithm that fetches data packets evenly and equally can utilize nodes' output bandwidth more efficiently, which can reduce operating costs of service provider. Finally, by analyzing solutions of optimization model, guidelines are proposed as principles of streaming algorithm design for real P2P streaming systems.

Key words: computer network; P2P streaming; network coding; streaming algorithm; streaming model

摘要: 在对等网(peer-to-peer,简称 P2P)流媒体系统中,节点(用户)的输出带宽(上行带宽)容量利用率的提高能够降低服务器的带宽开销.网络编码可以实现组播的最大吞吐率,因而具有提高系统中节点输出带宽容量利用率的潜力.将随机线性网络编码应用到 P2P 流媒体系统中,建立了基于随机线性网络编码的 P2P 流媒体传输过程模型,并据此建立传输算法的优化模型,比较研究了贪婪式算法、最少者优先算法和随机算法等.优化结果表明,随机算法可以平等均匀地获取数据包,能够最充分地利用节点的输出带宽容量,降低服务提供商的运营成本.通过对优化模型解的分析,对实际系统中的传输算法给出了设计指导原则.

关键词: 计算机网络;对等网流媒体;网络编码;传输算法;传输模型

* 基金项目: NEC 中国研究院资金

收稿时间: 2010-06-02; 定稿时间: 2011-01-20

CNKI 网络优先出版: 2011-05-26 13:54, <http://www.cnki.net/kcms/detail/11.2560.tp.20110526.1354.006.html>

中图法分类号: TP393

文献标识码: A

流媒体作为一种重要的网络应用,其用户规模迅速增长,所需的服务器带宽也急剧增加,传统的客户端-服务器(client-server,简称 C-S)模式已经无法满足需求.因此,如何降低服务器带宽开销成为网络流媒体系统急需解决的问题.由于对等网(peer-to-peer,简称 P2P)技术能够共享用户的存储、计算和带宽等资源,弥补服务器相应资源的不足,被研究人员应用到了流媒体传输,由此诞生了基于 P2P 技术的网络电视系统(P2P TV)^[1,2]和网络视频点播系统(peer-to-peer video-on-demand,简称 P2P VoD),并成功地实现了大规模商业应用^[3-5].

当前,由于各种原因,P2P 流媒体系统没能充分利用节点(用户)的输出带宽(上行带宽)容量.如新节点刚加入时,由于该节点没有可为其他节点服务的数据包(data packet),或者由于其他节点不能及时得到该节点加入的信息,导致该节点的输出带宽容量得不到充分利用;当节点退出后,该节点的邻居节点就会失去一个服务对象,可能会因此导致邻居节点的输出带宽容量得不到充分利用;低效的 P2P 拓扑构建算法和传输算法会使系统中出现一些节点需要获取数据包,但他们的邻居节点却没有各自所需的数据包,导致邻居节点的输出带宽容量得不到充分利用.

针对节点的输出带宽容量利用率低的问题,研究人员将网络编码应用到了 P2P 流媒体系统中.网络编码(network coding)技术^[6,7]可以实现组播的最大吞吐率,因而具有提高用户输出带宽容量利用率的潜力^[8].随机线性网络编码^[9]的提出,使得网络编码向实用化方向迈出了重要的一步.随机线性网络编码在 P2P 流媒体系统中的应用研究已经取得了一些进展^[8,10],但对于 P2P 流媒体系统核心之一的传输算法,仍然停留在使用启发式方法的阶段,系统性能无法预测.

本文对基于随机线性网络编码的 P2P 流媒体系统的传输过程建立了模型,并在其框架内对贪婪式算法、最少者优先算法和随机算法进行了比较研究,从而建立了算法优化模型并给出了求解方法.优化结果表明,随机算法,即平等均匀地获取数据包,能使得节点的输出带宽容量利用率最大化.通过对优化模型解的分析,本文为进一步设计实际系统中的传输算法给出了指导原则.

本文第 1 节介绍相关工作.第 2 节是对基于网络编码的 P2P 流媒体传输过程进行建模.第 3 节在此模型框架内研究传输算法,并对真实系统中的传输算法设计给出指导原则.

1 相关工作

无结构网状覆盖网(mesh overlay)技术解决了 P2P 流媒体系统所面临的节点动态性问题^[1],极大地提高了系统稳定性,但 P2P 系统中的节点输出带宽容量始终没能得到充分利用.网络编码可以实现组播的最大吞吐率,具有充分利用 P2P 流媒体系统中节点输出带宽容量的潜力.相同系统规模下,可以降低服务器输出带宽占整个系统输出带宽的比例,从而降低服务提供商的运营成本.

图 1 是 P2P 流媒体系统在相同状态下,使用网络编码前后的对比图.假设源节点的输出带宽容量为 3 个数据包/秒,其他节点输出带宽容量为 1 个数据包/秒.图 1(a)表示未使用网络编码的 P2P 流媒体系统的数据包传输情况.在当前系统状态下,节点 1~节点 3 都拥有相同的数据包 5~数据包 7,都需要获取数据包 8 和数据包 9,但此时只有源节点有这两个数据包,3 个节点只能从源节点获取,导致此时节点 1~节点 3 的输出带宽被浪费.图 1(b)中,基于网络编码的 P2P 流媒体系统不会出现这种情况.因为每个编码数据包都包含了段内 5 个原始数据包的信息,只要节点收到 5 个线性无关的编码数据包,就可以解出 5 个原始数据包.在图 1(b)的当前状态下,节点 1~节点 3 都有 3 个线性无关编码数据包,各自的 3 个编码数据包都包含了 5 个原始数据包的信息,对于各自的邻居节点来说都是有用的.每个节点可以从服务器获取数据包,也可以从源节点外的邻居节点获取数据包,避免了因各节点没有邻居节点所缺的数据包导致系统中节点带宽被浪费的情况,这使得服务器的输出带宽占整个系统的输出带宽比例由 100%下降到了 50%.

随机线性网络编码在 P2P 流媒体系统中的应用也带来一些挑战.例如数据包的大小如何决定,每个段内应包含多少个原始数据包,如何设计传输算法提高系统中节点的输出带宽容量利用率等.所以,研究人员做了一

些工作,探讨如何将随机线性网络编码应用到 P2P 流媒体系统.

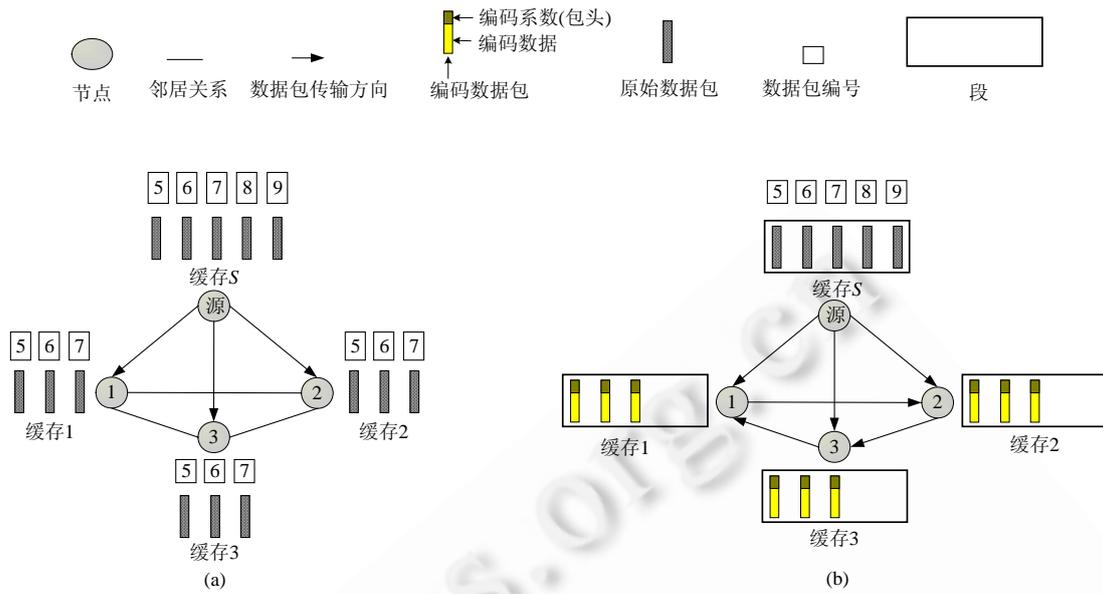


Fig.1 Illustration of comparison between P2P streaming systems with and without network coding

图 1 基于网络编码和未使用网络编码的 P2P 流媒体系统传输对比示意图

基于随机线性网络编码的 P2P 流媒体系统框图如图 2 所示,与传统的未使用网络编码的 P2P 流媒体系统相比,增加了网络编码和网络解码模块.网络编码模块用来对缓存中的数据包包进行编码生成新的编码数据包,为邻居节点服务.网络解码模块对收到的数据包进行解码.缓存管理模块用来对缓存进行插入、删除等基本维护操作,并读取缓存信息,生成缓存信息包向邻居节点广播.邻居管理模块用来寻找新的邻居节点,并维护邻居关系.数据传输模块决定本节点应该向哪些邻居节点请求自己所需要的数据包,或应该将数据包发送给哪个邻居节点.

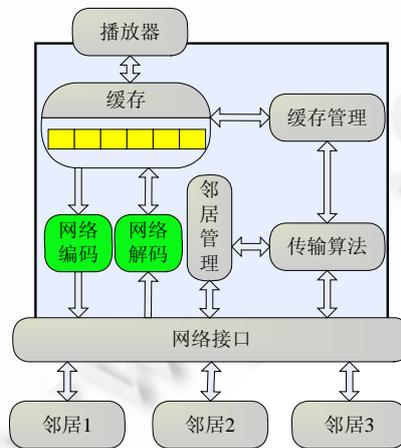


Fig.2 System diagram of network coding based P2P streaming

图 2 基于网络编码的对等网流媒体系统框图

Wang 等人^[8]将随机线性网络编码应用到 P2P 流媒体系统中,并通过计算机集群仿真实验表明,当系统整体

输出带宽容量略高于系统用户所需输入带宽(下行带宽)时,网络编码能够提高系统的吞吐率,改善用户的播放质量.Wang 等人^[10]基于文献[8]的经验,将数据传输方式从“拉(pull)”改为“推(push)”,并相应地调整了网络编码方式,Feng 等人^[11]进一步分析了该系统的理论性能.Liu 等人^[12]将 Wang^[10]和 Feng^[11]的研究成果应用到了商业 VoD 系统,结果表明,网络编码可以缩短用户的缓冲时间,提高用户输出带宽容量的利用率.

现有工作主要研究了随机线性网络编码的编解码复杂度、数据包大小的决定原则以及如何决定段内数据包数量等随机线性网络编码在应用中所面临的一些挑战,但对于作为 P2P 流媒体系统核心的数据传输算法部分都采用了启发式方法,缺乏理论分析.因为随机线性网络编码有一个特点,就是每个段在收齐段内原有数量的线性无关编码数据包时才能解出原始数据包,如果在系统需要播放该段时仍然没有收齐,则会导致已经收到的数据包被丢弃,传输这部分数据包的带宽被浪费掉.同时,P2P 流媒体系统中的节点可以有或无线方式接入,同一种方式的接入带宽也有差别,节点加入和退出的行为模式更是千差万别.所以,直观的启发式传输算法很难适应 P2P 流媒体系统的所有特点,导致节点的输出带宽得不到充分利用,这就要求研究人员对传输算法进行理论分析,为实际系统中的传输算法设计给出指导原则.

2 基于网络编码的 P2P 流媒体传输过程建模

P2P 流媒体的传输算法是传输过程的一部分,我们首先对传输过程进行建模,然后在此模型框架内对传输算法进行研究.

2.1 基于网络编码的P2P流媒体传输过程

基于网络编码的 P2P 流媒体系统在进行传输前,需要将连续的等长流媒体数据包划分为一系列的段,确定网络编码操作的范围.网络编码只能在段内操作,不能跨段.因为如果不进行分段,网络编解码模块在对第 1 个数据包编解码时需要所有的数据包,这会使得随机线性网络编码不能应用于持续不断播放节目的 P2P TV 系统.分段示意图如图 3 所示,该图中每 3 个等长的原始数据包划分为一个段.

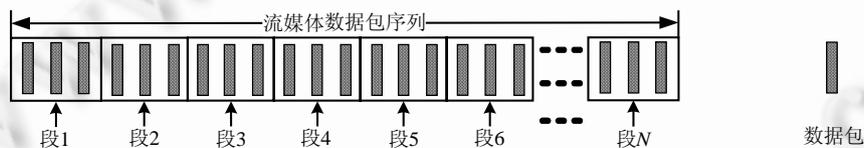


Fig.3 Illustration of segmentation for streaming data packets

图3 流媒体数据包分段示意图

在对流媒体数据包序列进行分段之后,基于网络编码的 P2P 流媒体系统就可以以段为单位,对流媒体数据包进行传输.

基于网络编码的 P2P 流媒体传输过程,是一个集数据包的编解码和传输于一体的过程.传输过程从源节点(服务器)开始,到各个目的节点(源节点外的其他节点,即用户)结束.各个目的节点也可以作为中间节点,利用自己的输出带宽为其他目的节点转发数据包.

传输过程如图 4 所示.源节点收到节点 1 的数据请求包后,通过网络编码模块对段内数据包进行编码,生成编码数据包,并传输给节点 1.节点 1 收到编码数据包后,将其存放到缓存中.当节点 1 的缓存中该段内的线性无关编码数据包达到两个以后,就可以作为中间节点为目的节点 3 服务,这两个节点之间的数据包传输过程与源节点和节点 1 之间的数据包传输过程一样.当任意目的节点收齐 3 个线性无关编码数据包后,就可以使用网络解码模块对收到的段内数据包进行解码,得到段内 3 个原始的数据包.例如,图 4 中节点 3 从节点 1 和节点 2 得到了 3 个线性无关编码数据包,并通过解码得到了 3 个原始数据包.节点 3 也可以像节点 1 和节点 2 一样作为中间节点为节点 4 和节点 5 服务.如果节点 3 收齐之后节点 2 还没有收齐,那么节点 3 也可以为节点 2 服务.

基于网络编码的 P2P 流媒体传输过程可以分为确定目的地、编码、传输、缓存和解码这 5 个步骤,下面分

别对这 5 个步骤的细节进行详细阐述。

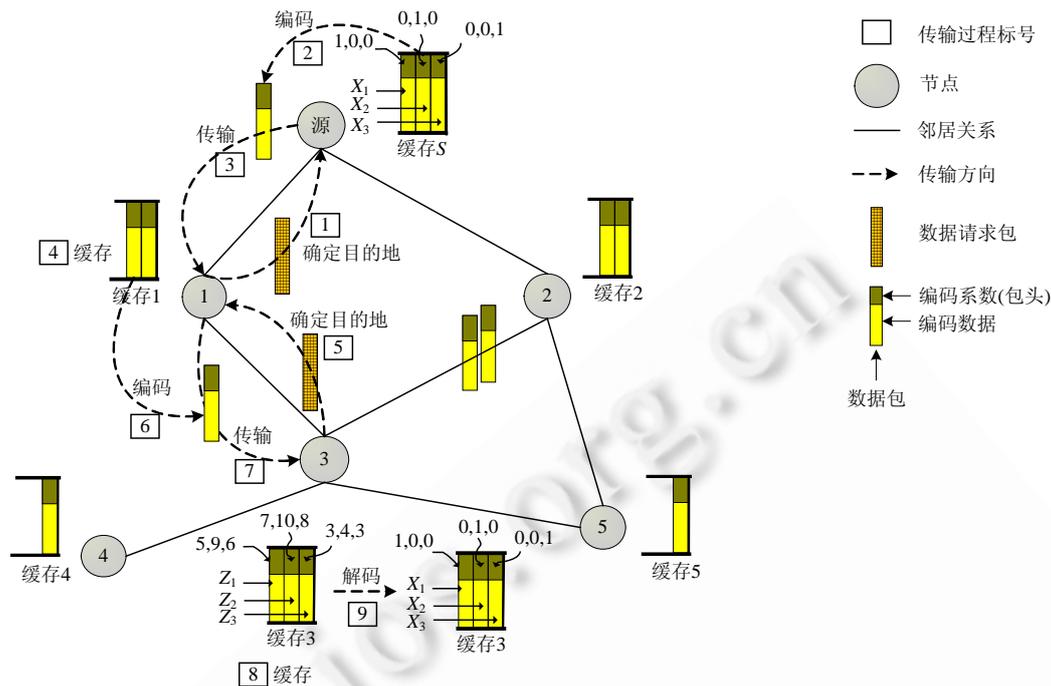


Fig.4 Illustration of P2P network coding streaming

图 4 P2P 网络编码传输过程示意图

2.1.1 确定目的地

如图 4 所示,对基于拉式传输算法的 P2P 流媒体系统来说,每个节点发送的数据包是由邻居节点主动请求的,也就是说,确定目的地这一步由数据包的接收节点决定,具体由传输算法模块完成。

传输算法在整个传输过程中处于核心地位,因为本文所讨论的 P2P 流媒体系统是基于无结构网状覆盖网,节点之间没有严格的父子关系,每个节点既可以将数据包传输给其他节点,也可以从相同的节点获得数据包.节点间的邻居关系也会因为节点的加入和退出、网络带宽、丢包率和时延的变化等不断调整.所以,每个节点需要周期性地根据自己的缓存状态决定该请求哪些段中的数据包包和相应段中的数据包包数量,并决定该向哪些邻居节点请求.我们用 $Q(\tau, n)$ 代表传输算法函数,表示某个节点 τ 时刻第 n 个段所请求的数据包包数量,并且从邻居节点中随机选择数据请求包的目的地节点.这里的 τ 时刻是指,节点的当前播放点为 τ ,即正在播放第 τ 个段。

2.1.2 编码

编码过程就是一个对向量进行线性组合生成新向量的过程,只不过这里的向量是段内的数据包.这一步由节点中的网络编码模块完成.编码过程如图 5 所示,当图 4 中的节点 1 为节点 3 服务时,首先随机生成一个编码向量,然后使用编码向量与已收到的该段中数据包进行线性组合,生成新的编码数据包和相应的编码系数。

为了避免数据溢出问题,随机线性网络编码的数学运算都在有限域**上进行.如图 6 所示,如果取有限域的范围是 2^8 ,则每个系数需要一个字节长度来表示.包含 3 个数据包的段,就需要 3 个字节来保存 3 个编码系数,并作为数据包的包头一起发给目的地节点.对于数据部分来说,也是把一个字节的内容看作一个符号(symbol),并分

** 又称伽罗瓦(Galois)域,即包含有限多个元素的域,这里指的是从 0 到 2^n-1 这 2^n 个数所组成的数域.对域内的元素进行这个域上所定义的加、减、乘和求逆运算之后的值仍属于这个域。

别与其他数据包相应位置的符号进行编码运算.如果有限域的范围取 2^{16} ,则两个字节作为一个系数或符号,对于包含同样数量数据包的段和同样长度的段中数据包来说,系数向量的长度会翻倍,数据部分的总长度不变,但所包含的符号数量会减半.

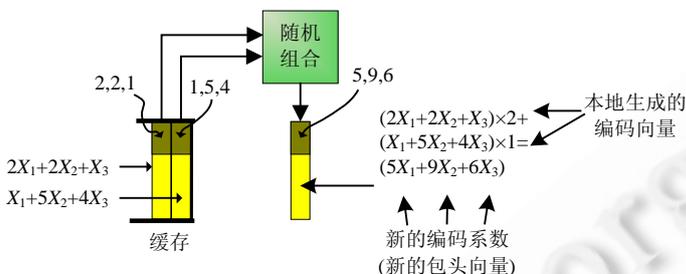


Fig.5 Illustration of encoding

图 5 编码示意图

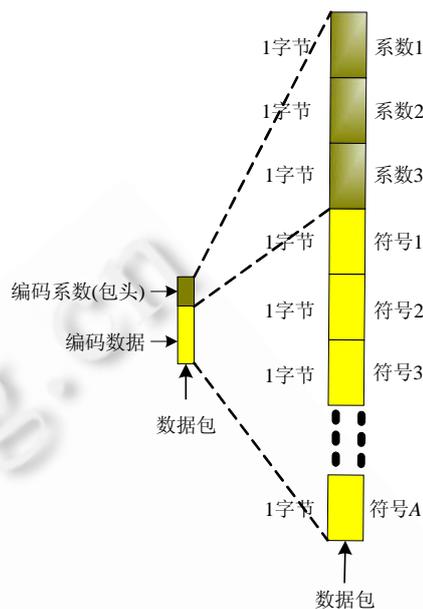


Fig.6 Illustration of data packet tagging

图 6 数据包标记示意图

对于源节点 S 来说,编码操作与其他节点一样,其缓存的各段中原始数据包可以看作特殊的编码数据包,编码系数矩阵是一个单位阵.

2.1.3 传输

图 4 中,节点 1 生成编码数据包后,使用 UDP 协议通过网络发送给目的节点 3.在数据包传输过程中,会由于源节点输出带宽容量、骨干网带宽容量和目的节点输入带宽容量的不足等原因产生丢包.对于 P2P 流媒体系统来说,节点的输出带宽容量是整个系统的瓶颈,骨干网带宽的容量通常都能够满足系统需求,输入带宽容量一般都高于流媒体码率.流媒体码率由服务提供商设置为低于大部分用户的输入带宽容量,以便争取到尽量多的用户.对于少部分输入带宽容量小于流媒体码率的节点,可能会因为不能容忍较差的流媒体质量而离开系统.

数据包传输过程的瓶颈主要由节点输出带宽容量的不足所引起,我们用丢包率 L 来表示这一瓶颈对数据包传输过程的影响,并假设丢包率为常数.如果某个段请求了 10 个数据包,丢包率为 0.4,则可以成功收到 6 个数据包.

节点的输入带宽容量也可能偶尔成为数据包传输的瓶颈.因为节点的输入速率由该节点所请求的数据包速率和丢包率决定,如果某个时刻请求的数据包过多,速率超过了节点的输入带宽容量,就会出现拥塞现象.为了避免数据包在节点输入处形成拥塞,浪费其他节点的输出带宽,节点在任何时刻的请求速率不能大于本节点的输入带宽容量,即

$$\sum_{n=\tau+1}^{\tau+B} Q(\tau, n) \leq CR, \forall \tau \tag{1}$$

这里假设节点的缓存长度为 B 个段,每个段所跨的时间长度为 1s.假设所有节点的输入带宽容量均为 CR ,其中, R 为流媒体码率, C 为节点输入带宽容量与流媒体码率的比值.

2.1.4 缓存

图 4 中,每个节点收到数据包后,都需要将数据包放到缓存中进行缓存.只要段中收到至少 2 个线性无关的编码数据包后就可以进行编码为其他节点服务.收齐段中原有数量的线性无关编码数据包之后,可以进行下一步的解码操作.

如果我们以 $H(\tau, n)$ 来表示第 n 个段在播放点为 τ 时所拥有的数据包数量,则 $H(\tau, n)$ 的取值应该等于第 n 个段在 τ 时刻之前所收到的数据包总和,即

$$H(\tau, n) = \sum_{r < \tau} [Q(r, n) \times (1 - L)], \forall n \quad (2)$$

在对某个段进行解码之前,该段必须已经收齐,即收到该段中原始数量的线性独立数据包.如果在该段到达播放点时仍没有收齐,则该段就被跳过,已经收到的数据包会被丢弃.这不仅会降低该节点的播放质量,也会浪费其他节点的输出带宽.所以,我们期望 $H(\tau=n, n)=M$ 或者 $H(\tau=n, n)=0$, 即每个段在到达播放点时能够收齐,或者一个数据包也没有收到.对于节点来说,只有 $H(\tau=n, n)=M$ 的那些段是有意义的,并期望这样的段越多越好.

我们引入状态标记函数 $J(n)$ 来表示第 n 个段到达播放点时是否已经收齐;引入函数 f 来表示整个流媒体数据包序列中,在到达播放点时能收齐的段个数.它们的表达式分别为

$$J(n) = \begin{cases} 1, & \text{如果 } H(\tau = n, n) = M \\ 0, & \text{其他} \end{cases} \quad (3)$$

$$f = \sum_n J(n) \quad (4)$$

2.1.5 解码

解码过程就是一个求解线性方程组解的过程.这一过程由节点中的网络解码模块完成,通常使用的是高斯-约当消去法.以图 4 中的节点 3 为例,当节点 3 收齐某个段中数据包后,就获得了数据矩阵 Z 和系数矩阵 A .其中,矩阵 Z 由 3 个编码数据包中的编码数据组成,矩阵 A 由 3 个编码数据包中的系数向量组成,且

$$Z = AX \quad (5)$$

$$\text{其中, } Z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}, A = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{bmatrix} = \begin{bmatrix} 5 & 9 & 6 \\ 7 & 10 & 8 \\ 3 & 4 & 3 \end{bmatrix}, X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

两边都左乘系数矩阵 A 的逆矩阵,就可以解出 3 个原始数据包 x_1, x_2 和 x_3 ,即

$$X = A^{-1}Z \quad (6)$$

2.2 数学模型

根据 P2P 流媒体网络编码传输过程的描述,我们可以得到传输过程的数学模型为

$$\begin{cases} \text{a) } \sum_{n=\tau+1}^{\tau+B} Q(\tau, n) \leq CR, \text{ 这里, } R = M, \forall \tau \\ \text{b) } H(\tau, n) = \left[\sum_{r < \tau} Q(r, n) \times (1 - L) \right] \\ \text{c) } J(n) = \begin{cases} 1, & \text{如果 } H(\tau = n, n) = M \\ 0, & \text{其他} \end{cases} \\ \text{d) } f = \sum_n J(n) \end{cases} \quad (7)$$

式(7)中,公式 a)表示本节点的输入带宽容量对传输算法的限制,每个请求周期内所请求的数据包速率不能大于本节点的输入带宽容量;公式 b)表示在播放点为 τ 时某个段 n 中的数据包数量,等于该段在 τ 时刻之前所收到的线性无关数据包总和;公式 c)表示具体某个段 n 在到达播放点时能否收齐,进而解码;公式 d)表示整个流媒体数据包序列中,某节点最终能收齐,然后解码并播放的段个数.

3 基于传输过程模型的传输算法研究

传输算法对流媒体传输过程起着决定性作用,并对流媒体质量产生直接影响.因此,如何设计传输算法,优化流媒体传输过程,是提高节点输出带宽容量利用率的关键所在.

各个 P2P 流媒体系统都设计了自己的启发式算法,而且千差万别.哪一种算法更适合基于随机线性网络编码的 P2P 流媒体系统,还有待进一步研究.如图 7 所示,缓存中每个段与播放点的距离不同,每个段在每个时刻所收到的数据包数量也会有所差异;同时,节点之间的邻居关系也在不断变化.这些复杂多变的因素要求 P2P 流媒体系统从理论上对传输算法进行分析,发现事物的本质,指导传输算法的设计.

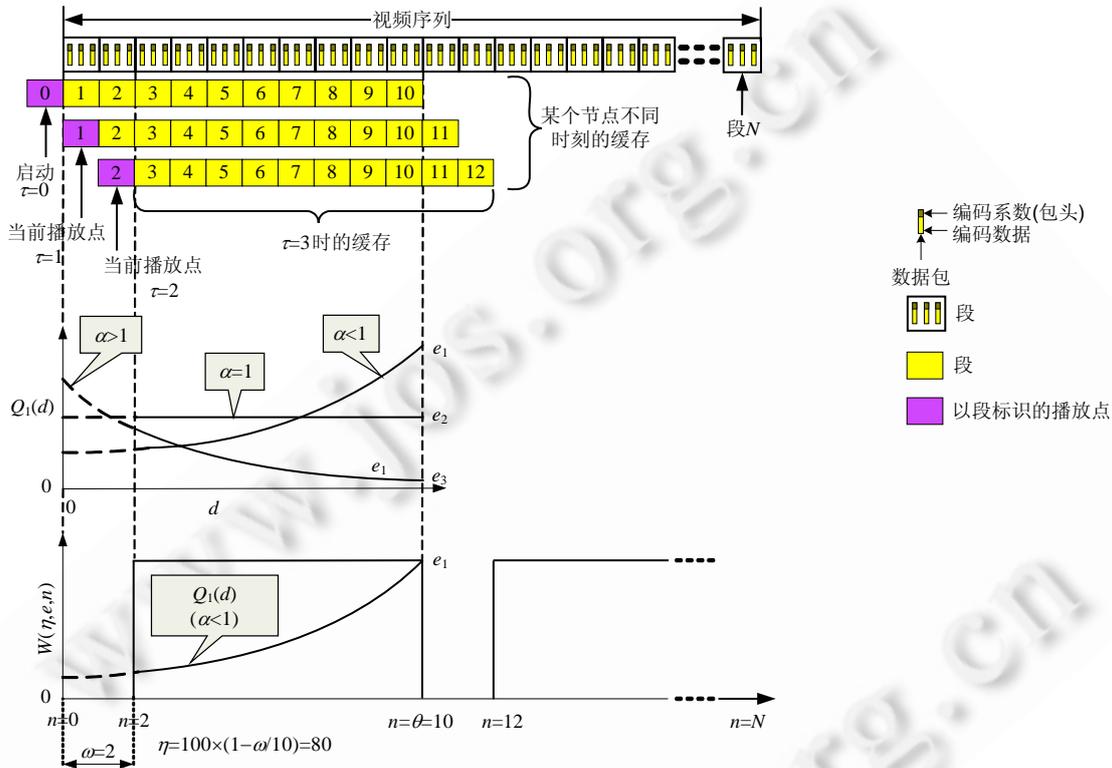


Fig.7 Illustration of streaming algorithm function

图 7 传输算法函数示意图

3.1 传输算法的数学描述

常用的传输算法一般有以下几种:

- 1) 贪婪式算法(greedy):是指首先请求缓存中离播放点最近的段;
 - 2) 最少者优先算法(rarest-first):是指首先请求在邻居节点中拥有者最少的缓存中的段.已有研究表明,这种算法倾向于首先请求缓存中离播放点较远的段^[13];
 - 3) 随机算法(random):是指随机地从待获取的段中选择一个.这种算法倾向于均匀获取每个段中的数据包.
- 我们设计了下面的传输算法,试图涵盖这些常用的算法:

$$Q_1(d) = Q_1(\tau, n) = \begin{cases} W \times \alpha^{B-d} = W \times \alpha^{B-(n-\tau)}, & \alpha \in [0, M], \text{ 如果 } d = n - \tau \in [1, B] \\ 0, & \text{其他} \end{cases} \quad (8)$$

Q_1 是一个指数函数,其形状如图 7 所示.当 $\alpha > 1$ 时,缓存中离播放点较近的段请求较多的数据包,类似于贪婪式算

法;但 $\alpha < 1$ 时,缓存中离播放点较远的段请求较多的数据包,类似于最少者优先算法;当 $\alpha = 1$ 时,缓存中各段平等获取数据包,与随机算法相似.

当节点可获得的输入速率低于流媒体码率时,肯定会出现一些段在到达播放点时无法收齐的情况.为了节省系统输出带宽,提高本节点的流媒体质量,我们期望无法收齐的段不获取任何数据包,而将可获得的数据包速率集中分配给一部分段.为了实现这一功能,我们将 Q_1 中指数函数的系数 W 改为一个方波函数,令

$$W(\eta, e, n) = e \times S\left(\eta, \frac{1}{\theta} \times (n - \omega)\right), \quad \eta \in [0, 100], e \in [0, M] \quad (9)$$

其中, $S(\eta, n)$ 是以 n 为自变量、周期为1、幅值为1、占空比为 η 的方波; ω 表示 W 的相位; θ 表示 W 的周期; e 表示 W 的幅值.并且为了简化,我们令

$$\omega = B - \frac{\eta \times B}{100} \quad (10)$$

$W(\eta, e, n)$ 的形状如图7所示.当 $W(\eta, e, n) = 0$ 时,第 n 个段就被放弃,不获取任何数据包;当 $W(\eta, e, n) > 0$ 时,第 n 个段按照传输算法 Q_1 获取数据包.

如果每个段按照传输算法 Q_1 获取数据包,可能会出现某些段获取的数据包多于它们所缺的数据包数量,造成数据包的冗余,浪费系统输出带宽,同时影响本节点获取所缺的其他数据包.为了避免出现这种情况,我们定义一个特殊的传输算法:

$$Q_2(\tau, n) = \begin{cases} M - H(\tau, n), & \text{如果 } n - \tau \in [1, B] \\ 0, & \text{其他} \end{cases} \quad (11)$$

即缓存中各段在每个请求周期内所请求的数据包数量,等于段中所缺的数据包数量.

实际的传输算法应该是符合 Q_1 ,同时受到 Q_2 的限制.综合 Q_1 和 Q_2 之后的结果如下:

$$Q(\tau, n) = \min(Q_1(\tau, n), Q_2(\tau, n)) \quad (12)$$

最终,传输算法的数学描述为

$$\begin{cases} \text{a) } Q(\eta, \alpha, e, \tau, n) = \min(Q_1(\eta, \alpha, e, \tau, n), Q_2(\eta, \alpha, e, \tau, n)) \\ \text{b) } Q_1(\eta, \alpha, e, \tau, n) = \begin{cases} W(\eta, e, n) \times \alpha^{B-d} = W(\eta, e, n) \times \alpha^{B-(n-\tau)}, & \alpha \in [0, M], \text{ 如果 } n - \tau \in [1, B] \\ 0, & \text{其他} \end{cases} \\ \text{c) } Q_2(\eta, \alpha, e, \tau, n) = \begin{cases} M - H(\eta, \alpha, e, \tau, n), & \text{如果 } n - \tau \in [1, B] \\ 0, & \text{其他} \end{cases} \\ \text{d) } W(\eta, e, n) = e \times S\left(\eta, \frac{1}{\theta} \times (n - \omega)\right), \quad \eta \in [0, 100], e \in [0, M] \\ \text{e) } \omega = B - \frac{\eta \times B}{100} \end{cases} \quad (13)$$

其中,对于 α 的取值范围说明如下:

- 1) $Q_1(\eta, \alpha, e, \tau, n) \geq 0$, 且 $W(\eta, e, n) \geq 0 \Rightarrow \alpha \geq 0$;
- 2) 每次请求数据包的数量不能大于段中原有数据包的数量,否则也会由于 Q_2 的限制变得没有意义,所以

$$Q_1(\eta, \alpha, e, \tau, n) \leq M \Rightarrow W(\eta, e, n) \times \alpha^{B-(n-\tau)} \leq M \quad (14)$$

因为对于需要获取数据包的段来说 $1 \leq W(\eta, e, n) \leq M$, 所以 $1 \leq \frac{M}{W(\eta, e, n)} \leq M$.

$$\Rightarrow \alpha^{B-(n-\tau)} \leq \frac{M}{W(\eta, e, n)} \leq M \quad (15)$$

其中, $1 \leq n - \tau \leq B$, 由此可得到 $0 \leq B - (n - \tau) \leq B - 1$. 当 $B - (n - \tau) = 0$ 时, 对 α 没有任何限制; 但 α 在 $B - (n - \tau)$ 的取值范围内都必须满足上述限制条件, 所以在 $0 \leq B - (n - \tau) \leq B - 1$ 的条件下, 即 $0 < \frac{1}{B - (n - \tau)} \leq 1$ 时, 有

$$\Rightarrow \alpha \leq M^{\frac{1}{B-(n-\tau)}} \leq M \quad (16)$$

3) 因此 $\alpha \in [0, M]$.

3.2 传输算法优化

在基于随机线性网络编码的 P2P 流媒体系统中,节点输出带宽容量一定的情况下,节点对整个流媒体序列能解码的段个数与邻居节点的输出带宽容量利用率成正比,同时前者也与节点的流媒体播放质量成正比,所以最大化任意节点的输出带宽容量利用率、最大化任意节点对整个流媒体序列的可解码段个数和最大化任意节点的流媒体播放质量三者等价.

3.2.1 传输算法的优化模型

将常用传输算法的数学描述代入 P2P 流媒体网络编码传输过程模型,以最大化任意节点的输出带宽容量利用率为目标,我们可以得到下面的优化模型.

目标函数

$$\text{Max}(f(\eta, \alpha, e)) \quad (17)$$

其中,

$$\left\{ \begin{array}{l} \text{a) } f(\eta, \alpha, e) = \sum_{n=B+1}^N J(\eta, \alpha, e, n) \\ \text{b) } J(\eta, \alpha, e, n) = \begin{cases} 1, & \text{如果 } H(\eta, \alpha, e, n, n) = M \\ 0, & \text{其他} \end{cases} \\ \text{c) } H(\eta, \alpha, e, \tau, n) = \left[\sum_{r=n-B}^{\tau} Q(\eta, \alpha, e, r, n) \times (1-L) \right] \\ \text{d) } Q(\eta, \alpha, e, \tau, n) = \min(Q_1(\eta, \alpha, e, \tau, n), Q_2(\eta, \alpha, e, \tau, n)) \\ \text{e) } Q_1(\eta, \alpha, e, \tau, n) = \begin{cases} W(\eta, e, n) \times \alpha^{B-(n-\tau)}, & \alpha \in [0, M], \text{ 如果 } n-\tau \in [1, B] \\ 0, & \text{其他} \end{cases} \\ \text{f) } Q_2(\eta, \alpha, e, \tau, n) = \begin{cases} M - H(\eta, \alpha, e, \tau, n), & \text{如果 } n-\tau \in [1, B] \\ 0, & \text{其他} \end{cases} \\ \text{g) } W(\eta, e, n) = e \times S \left(\eta, \frac{1}{\theta} \times (n - \omega) \right), \eta \in [0, 100], e \in [0, M] \\ \text{h) } \omega = B - \frac{\eta \times B}{100} \end{array} \right. \quad (18)$$

约束条件为

$$\text{i) } \sum_{n=\tau+1}^{\tau+B} Q(\eta, \alpha, e, \tau, n) \leq CR, R = M, \forall \tau \quad (19)$$

其中,公式 a)~公式 c)和公式 i)是传输过程模型,但排列顺序与传输过程正好相反;公式 d)~公式 h)是 3 种传输算法的数学描述.

因为第 1 个缓存长度的段没有足够的时间获取所需的数据包,所以为了排除其对整个结果的影响,在公式 a)中计算整个流媒体数据包序列的可解码段个数时,第 1 个缓存长度的段被排除在外.

该模型假设服务器的输出带宽容量为 R ,模型中的符号说明及常数设定见表 1.

Table 1 Definitions of symbols and functions

表 1 符号和函数定义

符号 (或函数)	说明	常数 取值
M	表示每个段中原始数据包的数量	30
R	表示系统流媒体码率	30
B	表示节点缓冲器中段的个数	20
n	表示段的标号.在整个流媒体播放过程中,每个段的标号是唯一的	
τ	表示当前播放点.以段号来标记	
L	表示丢包率	0.4
N	表示整个流媒体序列的长度.记为整个流媒体所包含的段的数量	200
C	表示节点的输入带宽容量与流媒体码率的比值	1.3
E	表示除源节点外每个节点的输出带宽容量与流媒体码率的比值	0.78
α	表示传输算法 Q_1 中指数函数的底.调节数据请求曲线的形状	
e	表示方波的幅值.即每个段进入缓存后,第 1 个请求周期内所请求数据包的个数,称为初始请求数据包数	
θ	表示方波的周期	20
ω	表示方波的相位	
η	表示方波的占空比	

注 1:速率和带宽容量均以数据包的数量来计量.

3.2.2 优化模型求解

因为本文所建模型中既有连续变量也有离散变量,所以在求解模型时采用了穷举的方式.需要求解的未知变量为 η, α 和 e , 取值范围分别是

$$\eta \in [0, 100], \alpha \in [0, 30], e \in [0, 30].$$

穷举的步长分别为 $100/B=100/20=5, 0.01$ 和 1 .

解出的最优解为 $f(\eta, \alpha, e)=117$, 即最终可播放的段的个数为 117 个, 其对应的变量取值如下:

- 1) 方波函数的占空比 $\eta=65$, 传输算法中指数函数的底 $\alpha=1.0$, 方波函数的幅值 $e=3$;
- 2) 方波函数的占空比 $\eta=65$, 传输算法中指数函数的底 $\alpha=0.99$, 方波函数的幅值 $e=3$;
- 3) 方波函数的占空比 $\eta=65$, 传输算法中指数函数的底 $\alpha=0.95$, 方波函数的幅值 $e=4$.

由上述最优解可知:

- a) 缓存中应该请求数据包的段个数由最优解 $\eta=65$ 决定, 为

$$B \times \eta / 100 = 20 \times 65 / 100 = 13 \quad (20)$$

并在到达播放点时能够收齐;

- b) 最优传输算法 Q_1 的函数形状是一条直线. 因为传输算法的函数形状由参数 α 决定, 其最优参数取值在 1.0 及其附近. 所以, 当某个段被包含在缓存中时, 最优的传输算法函数 $Q_1(\eta, \alpha, e, \tau, n)$ 的取值与 τ 和 n 无关, 每个请求周期内应该均匀地请求当前缓存中应该请求的那些段中的数据包;

- c) 每个应该请求数据包的段所获得的平均请求速率为每秒钟 $e=3$ 个或 $e=4$ 个数据包. 当 $\alpha=1.0$ 时 $e=3$, 缓存中每个应该请求数据包的段, 每个请求周期内平等地请求 3 个数据包. 此时, 参数 e 应该满足下面的不等式:

$$e \times (B \times \eta / 100) \leq C \times R \quad (21)$$

$$\Rightarrow e \leq \frac{C \times R}{(B \times \eta / 100)} = \frac{1.3 \times 30}{13} = 3 \quad (22)$$

$e=3$ 的最优解满足这个不等式, 并取等号.

3.3 优化结果分析

3.3.1 最优解分析

根据每个请求周期内的数据包请求速率的上限和丢包率, 可以计算出理想情况下节点可获得的速率为

$$C \times R \times (1 - L) = 1.3 \times 30 \times (1.0 - 0.4) = 23.4 \text{ (数据包/s)} \quad (23)$$

足以收齐缓存中段的个数为

$$[(23.4/M) \times B] = [(23.4/30) \times 20] = 15 \tag{24}$$

但模型中 η 的最优解小于 15, 等于 13, 即一个缓存长度的段中只有 13 个段可以收齐, 与最优值相差 2 个段. 这是因为本文研究的这类传输算法服从函数 $Q_1(\eta, \alpha, e, \tau, n)$ 的取值, 并满足每个段的上限限制 $Q_2(\eta, \alpha, e, \tau, n)$. 当某些段所缺的数据包少于 3 个时, 这些段所请求的数据包就会少于 3 个, 使得某些请求周期内所请求的数据包速率不能达到最大值 CR , 导致实际获得的输入带宽变小, 结果偏于保守.

由最优解可以计算出, 系统节点输出带宽容量的利用率为

$$\frac{13 \times M}{B \times E \times R} \times 100\% = \frac{13 \times 30}{20 \times 0.78 \times 30} \times 100\% = 83.3\% \tag{25}$$

3.3.2 解空间分析

解空间示意图如图 8 所示, 第 1 个最优解在图 8(e) 中用“*”标识. 每张图的 X 轴和 Y 轴分别是方波的幅值 e 和占空比 η , Z 轴是整个流媒体序列中可解码的段个数 $f(\eta, \alpha, e)$. α 的取值变化对解空间的影响由图 8(a)~图 8(i) 这一系列图的变化所展示出来.

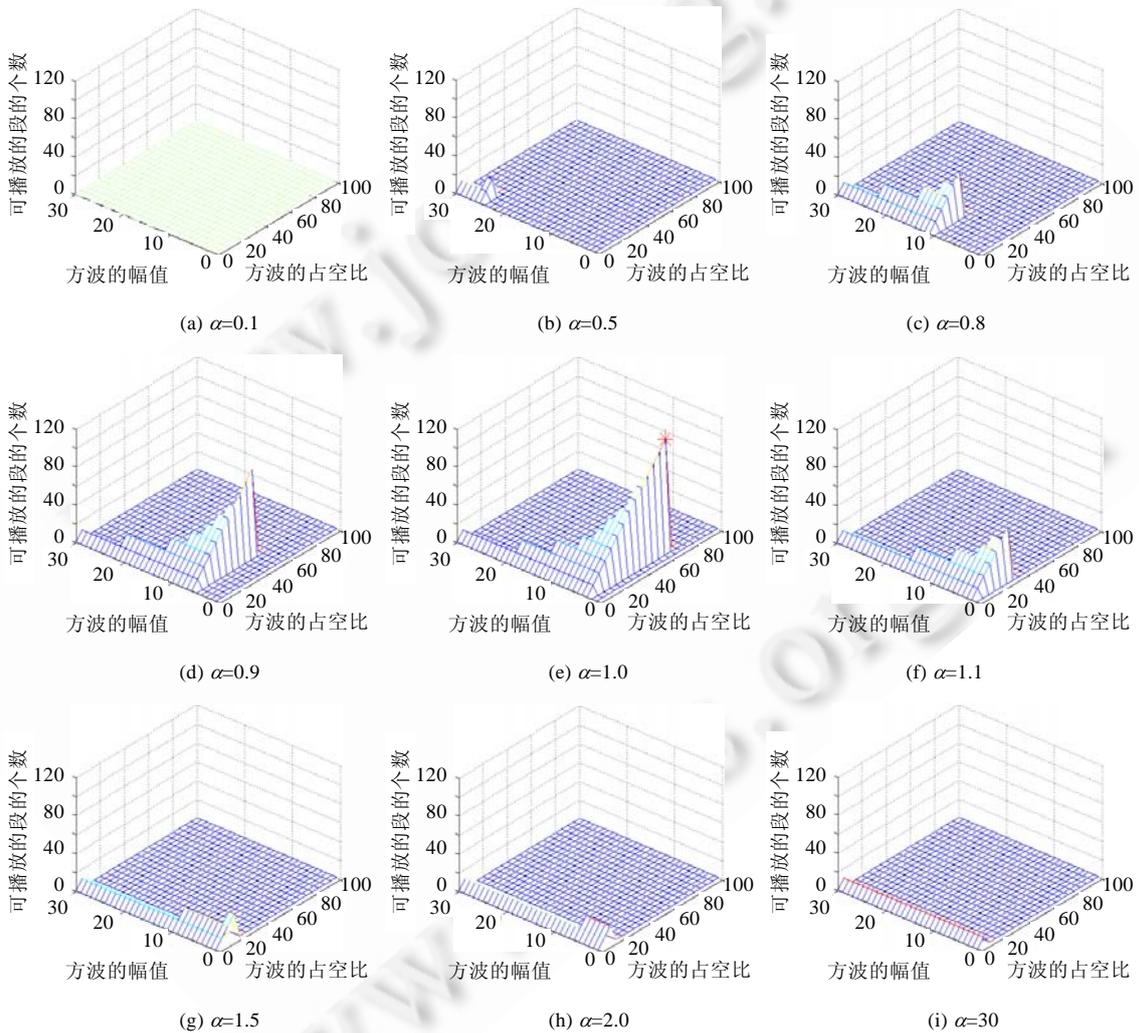


Fig.8 Illustration of solution space and optimal solution

图 8 解空间和最优解示意图

以图 8(e)为例, $f(\eta, \alpha, e)$ 取 0 时包含了两种情况:一种是所有的段到达播放点时都没有收齐;另一种是该点对应的 3 个参数 η, α 和 e 所决定的传输算法无法满足优化模型中的约束条件.如果方波的占空比 η 的取值过大,该节点可获得的数据包速率小于所有应该请求数据包的段收齐时所需的数据包速率.同时,传输算法试图将每个应该请求数据包的段都收齐,结果导致所有段到达播放点时都收到了相同个数的数据包,均小于段内原始数据包的个数 30,使得可解码段数为 0,即 $f(\eta, \alpha, e)=0$.如果方波的幅值 e 取值过小,会出现所有的段在 20 个请求周期内获取的数据包个数过少,导致所有段最终都无法收齐,使得 $f(\eta, \alpha, e)=0$.如果方波的幅值 e 取值过大,会使得传输算法 Q 无法满足优化模型中的约束条件,在计算过程中强制 $f(\eta, \alpha, e)=0$.

由图 8(a)~图 8(i)可以看出,以 $\alpha=1$ 为中心,随着 α 的增加或减小,整个流媒体序列中可解码的段个数都在减少.因为贪婪式算法类似于 $\alpha>1$ 时的情况,最少者优先方法类似于 $\alpha<1$ 时的情况,随机算法与 $\alpha=1$ 时的情况类似,所以随机算法与另外两种传输算法相比,更适合基于网络编码的 P2P 流媒体系统,更能充分利用系统中节点的输出带宽容量.随机线性网络编码要求每个段只有收齐才能解码的特点,决定了随机传输算法的优越性.因为贪婪式传输算法 Q_1 受到 Q_2 的限制,同时每次请求的数据包都有一定的丢包率,这使得每个段都需要经过多个请求周期才能收齐.当缓存中所缺数据包较多的段离播放点过近时,无法及时获取到所缺的数据包,导致最终无法收齐.越贪婪(α 增大),这种效应越明显,正如图 8(f)~图 8(i)所示.最少者优先算法正好与贪婪式算法相反,由于它过度重视离播放点较远的段,即每个请求周期内对刚进入缓存的段收取更多的数据包,而对离播放点较近、同时还没有收齐的那些段获取较少的数据包,导致一些段无法及时收齐,正如 α 从 1 开始减小到 0 的一系列图所展示的那样.随机算法取长补短,保留了贪婪式算法和最少者优先算法的优点,避免了两者的缺点,因此其性能最好.

3.4 传输算法的设计指导原则

本文的传输算法有 3 个变量 η, α 和 e, η 决定多少个段应该请求数据包, α 和 e 决定传输算法函数的形状和取值. η 的最优解由节点的可获得数据包的速率所决定. α 的最优解 1 决定了请求函数的形状是一条水平线,最优传输算法应该均匀地请求缓存中所有应该请求数据包的段中数据包.将可获得的数据包速率均匀地分配给应该请求数据包的段,由公式(21)和公式(22)可知,此时各段所得到的速率平均值就是模型中参数 e 的取值.也就是说,两个参数 η 和 α 的最优值确定之后,可以自然得到参数 e 的最优值.由此我们得到一些指导性原则,作为设计实际系统所用传输算法的依据,以充分利用节点的输出带宽容量,即:

- 1) 应该请求数据包的段的个数上限由公式 $[(r_d/M) \times B]$ 计算出来.其中, r_d 是节点近期输入速率的历史记录,方括号代表取整;
- 2) 平等均匀地请求应该请求数据包的段.

4 结 论

P2P 流媒体系统中的节点输出带宽容量利用率是系统性能的重要参数.节点输出带宽容量利用率的提高可以增加系统整体输出带宽,降低所需的服务器带宽开销,进而降低服务提供商的运营成本;或者在相同服务器带宽开销的情况下,可以支持更多的用户.本文将随机线性网络编码应用到了 P2P 流媒体系统中,对基于随机线性网络编码的 P2P 流媒体传输过程建立了模型.模型具有通用的框架,其参数具有明确的物理意义,体现了基于网络编码的 P2P 流媒体传输过程的特征,可以用来研究各种传输算法.在此模型框架内,本文建立了贪婪式算法、最少者优先算法和启发式算法的优化模型.优化结果表明,随机算法最能有效利用节点的输出带宽容量.

通过对模型求解的分析发现,应该请求数据包的段的个数上限是由节点的下载速率决定的,随机算法或平等均匀地请求所需数据包的段,可以更好地利用系统中节点的输出带宽容量.以此为指导原则设计的传输算法可以更好地利用 P2P 流媒体系统中的节点输出带宽容量,节省服务器带宽开销.

本文提出的传输算法优化模型对基于网络编码的 P2P 流媒体系统的部分传输算法进行了初步探索,进一步的研究主要围绕两个方面:一是将其他各种传输算法纳入到这个模型框架内,优化设计出更好的传输算法,以便进一步提高系统中节点的输出带宽容量利用率,降低服务提供商的运营成本;另一方面是将模型所给出的指

导原则应用于真实网络环境下的系统中,验证模型和算法的有效性.

References:

- [1] Zhang XY, Liu JC, Li B, Yum YSP. CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming. In: Znati T, ed. Proc. of the IEEE INFOCOM 2005. New York: IEEE Computer and Communications Society, 2005. 2102–2111. [doi: 10.1109/INFCOM.2005.1498486]
- [2] Zhang M, Luo JG, Zhao L, Yang SQ. A peer-to-peer network for live media streaming—Using a push-pull approach. In: Chua TS, ed. Proc. of the ACM Multimedia 2005. New York: ACM Press, 2005. 287–290. [doi: 10.1145/1101149.1101206]
- [3] Huang Y, Fu TZJ, Chiu DM, Lui JCS, Huang C. Challenges, design and analysis of a large-scale P2P-VoD system. In: Bahl V, ed. Proc. of the ACM SIGCOMM 2008. New York: ACM Press, 2008. 375–388. [doi: 10.1145/1402958.1403001]
- [4] PPLive. <http://www.pplive.com/en/>
- [5] PPStream. <http://www.ppstream.com/>
- [6] Ahlswede R, Cai N, Li SYR, Yeung RW. Network information flow. IEEE Trans. on Information Theory, 2000,46(4):1204–1216. [doi: 10.1109/18.850663]
- [7] Chou PA, Wu Y. Network coding for the Internet and wireless networks. IEEE Signal Processing Magazine, 2007,24(5):77–85. [doi: 10.1109/MSP.2007.904818]
- [8] Wang M, Li BC. Lava: A reality check of network coding in peer-to-peer live streaming. In: Baldwin RL, ed. Proc. of the IEEE INFOCOM 2007. New York: IEEE Computer and Communications Society, 2007. 1082–1090. [doi: 10.1109/INFCOM.2007.130]
- [9] Ho T, Koetter R, Médard M M, Karger DR, Effros M. The benefits of coding over routing in a randomized setting. In: Imai H, ed. Proc. of the IEEE Int'l Symp. on Information Theory 2003. New York: IEEE Information Theory Society of the Institute of Electrical and Electronics Engineers, 2003. 442–442. [doi: 10.1109/ISIT.2003.1228459]
- [10] Wang M, Li BC. R^2 : Random push with random network coding in live peer-to-peer streaming. IEEE Journal on Selected Areas in Communications, Special Issue on Advances in Peer-to-Peer Streaming Systems, 2007,25(9):1655–1666. [doi: 10.1109/JSAC.2007.071205]
- [11] Feng C, Li BC. On large-scale peer-to-peer streaming systems with network coding. In: Saddik AEL, ed. Proc. of the ACM Multimedia. New York: ACM Press, 2008. 269–278. [doi: 10.1145/1459359.1459396]
- [12] Liu ZM, Wu C, Li BC, Zhao SQ. UUSee: Large-Scale operational on-demand streaming with random network coding. In: Mandyam G, ed. Proc. of the IEEE INFOCOM 2010. New York: IEEE Computer and Communications Society, 2010. 1–9. [doi: 10.1109/INFCOM.2010.5462030]
- [13] Zhou YP, Chiu DM, Lui JCS. A simple model for analyzing P2P streaming protocols. In: Wu JP, ed. Proc. of the IEEE Int'l Conf. on Network Protocols 2007. New York: IEEE Press, 2007. 226–235. [doi: 10.1109/ICNP.2007.4375853]



张志明(1979—),男,河南博爱人,博士生,主要研究领域为P2P流媒体.



周晋(1977—),男,博士,主要研究领域为结构化P2P系统,P2P流媒体.



陈震(1976—),男,博士,讲师,主要研究领域为高速网络,P2P系统,可信计算.



李军(1962—),男,博士,研究员,博士生导师,主要研究领域为网络安全,网络体系结构.