基于 P2P 的视频点播系统综述

沈时军 李三立
(清华信息科学与技术国家实验室 北京 100084)
(清华大学计算机科学与技术系 北京 100084)

摘 要 近十年来,基于对等网络(Peer-to-Peer,P2P)的视频点播系统(Video-on-Demand,VoD)受到了越来越多的关注.它吸引人的原因在于,与传统的基于服务器/客户机结构的视频点播系统相比,P2P 技术具有成本低、扩展性好的优点.但是,由于对等网络内在的不稳定性、异构性,这类系统在实现上面临着诸多挑战.文中对现有的该类系统的体系结构进行模块划分,并对各模块的实现策略进行讨论;特别是对 VoD/P2P 实现中的 3 个主要方面,即数据传输、数据存储、激励机制进行了综述.

关键词 对等网络;视频点播;传输拓扑;编码方案
中图法分类号 TP393 **DOI 号**: 10.3724/SP.J.1016.2010.00613

P2P-Based Video-on-Demand Systems: A Survey

SHEN Shi-Jun LI San-Li
(*Tsinghua National Laboratory for Information Science & Technology, Beijing* 100084)
(*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)

Abstract Over the past decade, peer-to-peer (P2P) based video-on-demand (VoD) systems have received a tremendous amount of attention. They are immensely attractive because of their cheap deployment and high scalability compared to traditional Client/Server (C/S) based VoD systems. However, due to the instability and heterogeneity of P2P networks, there are many challenges in designing, implementing, and deploying such systems. For instance, the lack of resources (e.g., source nodes, bandwidth, storage, etc.) usually becomes a barrier that prevents practical systems from normal operation. Based on the observation, this paper outlines the components of existing VoD/P2P architectures and surveys approaches to their design, focusing on how to make full use of limited resources in P2P based VoD systems. Specifically, the authors present a survey in three aspects: (1) data transmission, referring to how to make full use of nodes' bandwidth; (2) data storage, referring to how to make full use of nodes' storage; and (3) incentive mechanisms that stimulate nodes to contribute more resources.

Keywords Peer-to-Peer; video-on-demand; transmission topology; encoding scheme

1 引 言

20 世纪 90 年代,在万维网取得了巨大成功之后,人们进一步希望可以在网上随意观看自己喜爱的视频节目,即视频点播.最初的 VoD 系统是基于 C/S 结构的,为了增加系统支持的用户量,并降低用户的等待时间,人们提出了 batching[1-2]、patching[3]

收稿日期:2008-09-01;最终修改稿收到日期:2010-03-05.本课题得到国家精品课程集成项目(JPKC-5)资助.**沈时军**,男,1982 年生,博士研究生,主要研究方向为流媒体技术、移动计算.E-mail:csj@mails.tsinghua.edu.cn.**李三立**,男,1935 年生,教授,博士生导师,中国工程院院士,主要研究领域为高性能集群计算、网格计算、移动计算、流媒体技术等.

等一系列算法.但点播不同于传统的 Web 应用,其高带宽消耗、时延敏感的特性很快使 C/S 结构暴露出可扩展性差的问题.于是,一个自然而然的想法是将多个服务器分散部署到网络中,当用户点播节目时,可以从距离自己最近的服务器取得数据.这就是 CDN(Content Distribution Networks)的思想,它一定程度上提高了点播系统的可扩展性,但同时引入了新的问题:由于需要很多服务器,整个系统的部署与维护将非常昂贵.对于这个问题,人们很快有了解决办法:他们发现用户终端的性能不断提高,甚至可以达到服务器的标准.由此,基于对等网络的视频点播系统(下文称为 VoD/P2P)出现了.视频点播并不是采用 P2P 技术的最初应用,事实上在它之前,基于 P2P 的文件共享系统已经非常流行.第一个现代意义上的 P2P 文件共享系统开始于 1999 年,一个年仅 18 岁的美国人开发了 Napster,并在它上面共享音乐文件.Napster 迅速流行,在 2001 年达到了 150 万最高在线用户.虽然 Napster 最终因为版权问题而停止运行,但它引发了 P2P 使用的高潮.在它之后很多类似的系统,如 Gnutella、KaZaA、eDonkey、Bittorrent 等,今天仍被广泛使用.在文件共享系统之后,P2P 的另一个重要应用是视频直播.大约从 2004 年底到 2005 年,基于 P2P 的网络视频渐渐流行起来,如 Coolstreaming、GridMedia、PPLive、PPStream 等.这些系统通过转播体育娱乐节目,吸引了大量用户.到了 2007 年,P2P 直播已经取得了相当成功,几乎没有人怀疑它的广阔前景.也就在这时,很多公司,包括 PPStream、PPLive、UUSee、VeohTV、Joost,推出了另一个更具挑战性的业务——P2P 点播.

文件共享、视频直播、视频点播有一个共同的特点:耗带宽.为了提高系统的可扩展性并降低成本,P2P 技术成为它们的首选.但是,采用 P2P 技术又有一些不利的因素.例如,P2P 网络是动态的、不稳定的(churning),节点的上线、下线是任意的.如果正在提供服务的节点突然下线了,将会对接收服务的节点造成影响.而且,P2P 网络中节点的差异性很大,如上传带宽,有一些校园网用户可以达到 100Mbps,但某些 ADSL 用户可能只有几百 Kbps,甚至达不到一个视频的平均码率.这些因素对文件共享系统来说,可能只是影响文件的下载时间;但对时延敏感的视频直播、点播应用,却可能造成播放不流畅,甚至无法播放的严重后果.

视频的直播与点播,虽然相似,但又有很多不同之处.直播更像电视,用户只能选择看或不看,并没有太多的交互性;点播则更像 DVD,用户可以选择何时播放,并且在观看过程中可以进行暂停、恢复、拖动播放等 VCR 操作.因此,同一频道不同用户的播放进度,在直播中是相近的,而在点播中却可能相差很大.播放进度差异的直接影响是,用户间数据共享的机会更少.因此在实现上,点播比直播更具挑战性.表 1 给出了基于 P2P 的视频直播与视频点播的一个比较.

表 1 基于 P2P 的视频直播与视频点播的比较

	视频直播	视频点播
请求并发性	请求并发性高	请求异步、分散
端到端延迟	希望尽可能小	一般没有要求
频道数量	一般较少,用户集中于少量频道	一般很多,用户分散在不同频道
VCR 要求	实时节目,交互性少	一般支持暂停、恢复、拖动播放等操作
数据共享难度	用户间播放进度差异小,共享容易	用户间播放进度差异大,共享困难

面对高带宽、高存储、高实时要求的 VoD 应用,P2P 网络中有限的资源可用性的问题显得非常突出.这表现在两个方面:(1)由于节点硬件条件(带宽、存储)的限制,能提供的资源是有限的;(2)由于缺乏一个有效的激励机制,节点愿提供的资源是有限的.目前,虽然已有很多大型的 VoD/P2P 系统,如 PPLive、PPStream、迅雷等,但它们在很大程度上依赖于服务器或 CDN,因而普遍存在“成本过高、盈利困难”的问题.本文致力于研究如何在 VoD 应用中发挥 P2P 技术的优势,对 VoD/P2P 系统在 3 个方面作综述,也就是:(1)数据传输.讨论如何充分地利用节点的有限带宽;(2)数据存储.讨论如何充分地利用节点的有限存储;(3)激励机制.讨论如何刺激节点贡献资源.

本文第 2 节讨论 VoD/P2P 的数据传输,包括传输过程中的拓扑结构以及编码技术;第 3 节讨论 VoD/P2P 的数据存储,包括存储过程中的资源查找结构、编码方案以及替换策略;第 4 节讨论 VoD/P2P 的激励机制,介绍 3 种常见的策略;最后,第 5 节总结全文.

2 传输问题

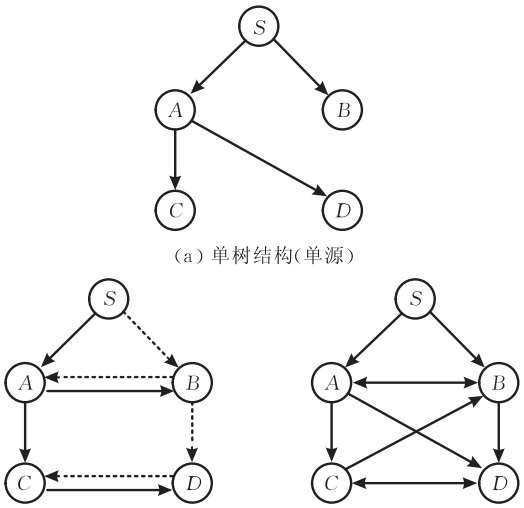
VoD/P2P 的传输希望解决如下的问题:如何从一个或多个源节点向多个目标节点分发数据,使之在动态的、异构的网络中:(1)保证视频的传输率,

即目标节点的有效输入带宽不能小于视频的码率；(2)减少用户等待时间，即用户从发出点播请求到视频播放的时间间隔尽可能短；(3)降低带宽浪费，即点播不能消耗很多不必要的带宽而影响其它应用. 近年来，对传输问题的研究主要集中在拓扑结构与编码技术两方面.

2.1　传输中的拓扑结构

在早期的基于 C/S 结构的点播系统中，人们曾希望采用 IP 组播技术来降低系统的带宽消耗. 但遗憾的是，在实际环境中，IP 组播并不能很好地被支持. 不过借助于 P2P，IP 组播可以有一个替代策略，即应用层组播(Application Layer Multicast，ALM). 它的基本思想是，在应用层，P2P 网络的每个节点都充当路由器的角色，将收到的数据转发给其它多个节点. 这其实是在应用层构建了另一个“网络层”，或称为 Peer Overlay. 本文接下来所说的传输中的拓扑结构，也就是 Peer Overlay 的拓扑结构.

在 Peer Overlay 中，若每个节点最多只有一个上游节点提供数据，称这样的拓扑为单源结构. 而与此对应的，若每个节点可以有多个上游节点提供数据，称这个拓扑为多源结构. 对单源、多源结构的一个更严格的定义是：P2P 网络中，针对某一个频道，将其中的节点抽象为有向图的点，将一个节点向另一个节点提供数据的链路抽象为有向图的边，则形成一个有向图 G；如果 G 中每个点的入度不大于 1，称 G 是单源结构，反之，称 G 为多源结构. 图 1 给出了不同拓扑的 Peer Overlay 的例子.



(a) 单树结构(单源)

(b) 多树结构(多源)，实线与虚线代表两棵不同的组播树

(c) 网状结构(多源)

图 1　传输中的拓扑结构(图中箭头表示数据流向)

在 VoD/P2P 的早期研究中，大多数的传输拓扑是单源结构的，如树状[4-9]、链状[10]、环状[11]. 在 P2Cast[9] 中，同一频道中到达时间相近的 peer 组成一个 session. 在每个 session 中，server 连同这些 peer 构成一棵应用层的组播树. 那些到达时间稍晚的 peer，也可以加入到某一个 session 中，并从这个 session 的 server 或 peer 取得它所错过的数据(即补丁流). 在这个系统中，peer 的主要作用是：(1)作为组播树的节点转发数据；(2)为后继的 peer 提供补丁流. 在文献[10]中，为了避免上游节点由于带宽有限而成为系统瓶颈，作者提出了一个父子交换协议(Parent Child Exchange，PCX)，通过某种规则动态地交换上、下游节点，以优化拓扑结构. 为了能快速修复上游节点的失效，P2VoD[8] 将到达时间相近的节点(播放进度也相近)组织在一起，称为一个 Generation. 当某节点的上游节点失效后，该失效节点所在的 Generation 中任何一个其它节点都可以替代失效节点，成为新的上游节点. 类似的思想也出现在文献[6-7]中.

单源结构有很多优点. 首先，它很直观，从而很容易对系统进行调试和评估；其次，当系统稳定时，上游节点一般知道下游节点需要什么数据，从而可以通过推模式(push-based)向下游提供数据，减小端到端延迟；最后，因为拓扑简单，单源结构可以理论上优化，以达到最佳的系统性能. 但是，单源结构又有很多不足，表现在：(1)资源利用率低. 只有源节点与中间节点提供数据，而大量的叶子节点并不提供数据，它们的资源(主要是上传带宽)并没有得到有效利用. 这导致的另一个后果是负载不均衡；(2)对服务节点要求太高. 单源结构中，一个节点要提供服务，它的上传带宽必须大于视频码率.

由于单源结构的缺点，多源结构逐渐流行起来. 其思想是：将原始视频转换成很多数据块，一个节点同时从若干个不同的上游节点取得不同的数据块，当收集到足够数量的块时，就可以恢复出原始视频. 多源结构可以有效地避免单源的不足：由于节点间相互交换数据块，每个节点都可以贡献资源，资源利用率更高；通过将视频分成很小粒度的数据块，也降低了节点能提供服务的门槛.

从本质上说，多源结构由于节点间相互提供数据，没有明确的上游与下游，其拓扑必然是网状的. 但是从设计的角度，它可以分为多树结构[12-13]、网状结构[14-22]、混合结构[23-24]等. 多树结构通常由多棵组播树组成，每一棵子树对应一个视频子流；而网状结构则没有“树”的任何特征，视频数据甚至不表现为具体的流，而是以块的形式传输. 图 1(b)给出了

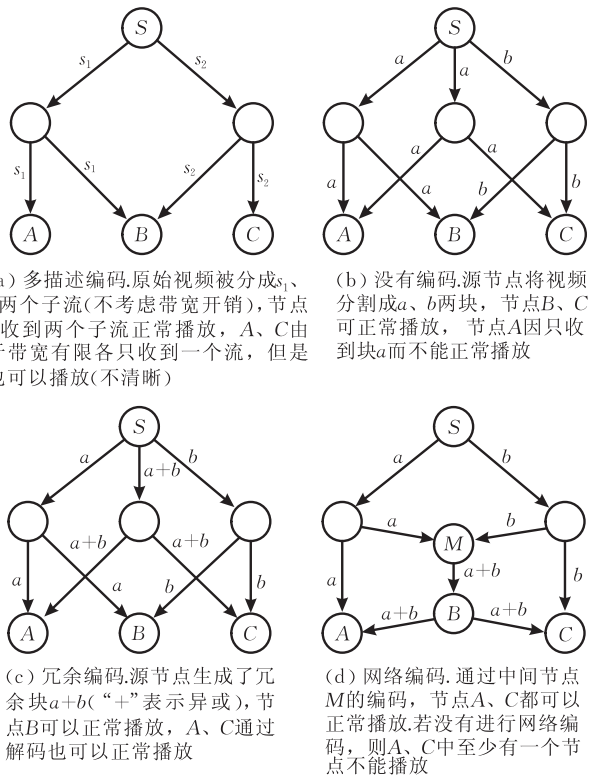
多树结构的一个例子,其中的视频被分成两个子流.由于需要维护多棵组播树,多树结构的实现较复杂,反而是网状结构由于实现简单却同样有效,近年来更加流行.Coolstreaming[18]是一个典型的网状结构.在Coolstreaming中,视频按照播放顺序被分割成一系列数据块.每个节点维护若干个与自己播放进度相近的节点,并周期性地与它们交换信息,当发现其它节点有自己没有的数据块时,就从这些节点请求数据.在文献[22]中,作者提出了一个共同体(alliances)的概念.一个共同体由若干个节点组成,一个节点可以参加若干个共同体.在共同体内,按照互利的原则,数据是协同下载与共享的.这个由共同体组成的网状结构,可以抽象为小世界网络,从而具有小世界网络的某些优点.作者认为这样的一个系统有更好的可扩展性,比Coolstreaming有更好的QoS保证.

多源结构比单源结构有更大的优势,但是在采用多源结构以前,有一个问题需要考虑:由于存在多个上游节点,收到的数据很容易造成重复,从而浪费带宽.通过协商可以一定程度上减少数据重复,但由于实际环境中协商并不总是及时的,数据重复的情况并不能避免;另一方面,协商本身引入了通信开销与延迟,而且在协商过程中,带宽并不能充分地利用.幸运的是,还有另一种处理数据重复的办法——编码.本文接下来讨论传输中的编码.

2.2 传输中的编码

在VoD/P2P的数据传输过程中,编码的运用主要有如下目的:(1)增强网络容错性.当网络丢包或上游节点失效时,不对当前节点产生影响,或影响很小;(2)提高数据差异性.即便在没有协商的情况下,从不同上游节点接收的数据也没有重复,或者产生重复的概率很小;(3)适应节点异构性.将原始的视频流编码成若干子流,每个子流的带宽要求较低,所以那些下载带宽较小的节点若没有能力接收整个视频流,可以接收少量的子流,同样可以播放视频(清晰度降低).本文接下来介绍3种主要的编码思想(图2),即多描述编码(Multiple Description Coding,MDC)、冗余编码、网络编码(Network Coding).

多描述编码是为了适应网络丢包,并适应不同带宽用户而提出的一种编码思想.它将原始的视频流编码成若干个子流,取得任意数量的子流都可解码播放,并且取得的流越多,播放质量越高.编码的具体实现有多种,可以是基于视频的时域分割,也可以是基于视频分层编码,或者是一些其它的设计[25].多描述编码有很好的容错性,因此它在很多系统中被采用[12-13,26-27].但它有一个很大的缺点,就是当编码的子流数目少时,并不能表现它的优势;而当子流数目多时,引入的带宽开销很大.有研究表明[28],当编码成2个子流,引入的带宽开销大致在10%~50%;而当编码成8个流时,开销在20%~200%.具体的开销与编码实现及视频内容有很大的关系.



(a)多描述编码.原始视频被分成s_1、s_2两个子流(不考虑带宽开销),节点B收到两个子流正常播放,A、C由于带宽有限各只收到一个流,但是也可以播放(不清晰)

(b)没有编码.源节点将视频分割成a、b两块,节点B、C可正常播放,节点A因只收到块a而不能正常播放

(c)冗余编码.源节点生成了冗余块$a+b$("+"表示异或),节点B可以正常播放,A、C通过解码也可以正常播放

(d)网络编码.通过中间节点M的编码,节点A、C都可以正常播放.若没有进行网络编码,则A、C中至少有一个节点不能播放

图2 传输中的不同编码方案

(图中箭头表示数据流向,S表示源节点)

与多描述编码相比,冗余编码和网络编码引入的带宽开销则小得多,一般只是编码中用到的一些元数据信息.事实上,冗余编码和网络编码拥有相同的算法基础,即或者基于数论中的有限域,或者基于图论中的稀疏二部图.为了适应点播的"边下载边播放"的特点,在编码前,视频文件首先按播放顺序被分割成若干段(比如每1段有1秒的播放时间),称之为原始数据段.基于有限域的编码流程是:对每一个原始数据段,将其平分成N块,然后将这N数据块在$GF(2^k)$域上作线性叠加以生成冗余块.当GF域足够大时,生成的冗余块间线性相关的概率很小,所以任取N(或稍大于N)冗余块即可解码出原始数据段.例如,Reed Solomon Code是一个常用的基于有限域的编码.基于稀疏二部图的编码流程是:对每一个原始数据段,将其平分成N块,然后按照某

种分布从中任选 r 块($r \leq N$)，并将这 r 数据块作“异或”操作以生成冗余块. 为了使任取 N(或稍大于 N)冗余块即可解码，实际的编码过程更复杂一些. 相关的编码实例包括 Tornado Code[29]、LT Code[30]、Raptor Code[31]等. 如果一个编码可以从原始块中生成无穷多的冗余块，则称这个编码为 rateless code，反之则称为 rated code.

冗余编码与网络编码的主要区别在于，前者是离线编码. 具体地说，冗余编码是在数据传输前(离线状态下)，将原始数据段分成 N 块，并利用某种算法(如 Reed Solomon Code)扩展出 M 块冗余，从这 $N+M$ 数据块中任取不同的 N 块就可解码出原始数据. 冗余编码其实更多地针对存储而非传输[32]，因为它的编码过程发生在传输前，而且它的优势主要表现在存储中(参见第 3 节). 但在传输过程中，同样可以因为它的“冗余”而得到好处. 图 2(b)、(c)对冗余编码与没有编码的方案作了比较. 冗余编码的一个缺点是，当冗余块数 M 较小时，在没有协商的情况下，任取 N 数据块，因发生重复而不能解码的概率仍较大(如图 3 所示). 例如 $N=6$，$M=4$ 时，这个概率是 85%；即使 $M=1000$，这个概率仍有 1.5%. 而另一方面，当 M 值很大时，按照图 2(c)的结构，源节点需要预先缓存很多冗余块，将消耗大量存储空间. 当然，后一个问题可以通过一边编码一边发布的形式解决，其实这已经具有了网络编码的某些特征.

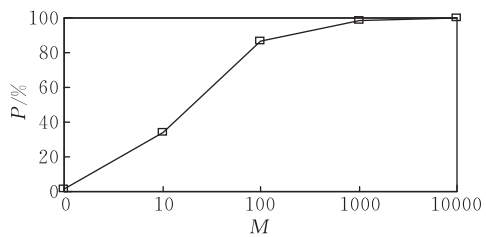


图 3　冗余编码的数据可用性. 采用 Reed Solomon 编码，将原始视频分成 $N=6$ 块，然后扩展出 M 个冗余块。横坐标表示 M 值，纵坐标表示当网络中存在 N 个随机的数据块时，可以解码出原始视频的概率 P. $M=0$ 表示没有编码

网络编码刚产生时[33]，就被认为是一种很有前景的编码. 它的思想是：当一个源节点向多个目标节点组播数据时，中间节点在转发收到的数据之前先进行编码[34-37]. 网络编码可以让网络达到最大流的吞吐量. 图 2(d)给出了网络编码的一个例子. 在这个例子中，如果不采用网络编码，A、C 两节点至多只有一个节点能播放视频. Avalanche[35]是一个采用网络编码的大型内容分发网络. 实验表明，该系统文件下载时间比没有编码的系统缩短了 2～3 倍，并且采用网络编码提高了系统的鲁棒性. 在 rStream[34]，由于采用基于 rateless code 的网络编码，每个节点理论上只要能支持与原始视频码率相当的下载速率即可播放，而不用担心收到的数据会重复，因此系统主要将精力放到如何最小化端到端的延迟上. 在 Lava[36]中，作者对直播/点播系统中网络编码的作用作了系统的研究，得出的结论为：(1) 网络编码可以使流调度更加细粒度化，并减少数据重复和带宽消耗；(2) 网络编码可以更好地适应网络动态性；(3) 对于物理带宽刚刚能满足视频码率的用户，网络编码可能是最有效的手段.

在文献[38]中，作者对基于 Reed Solomon Code 的网络编码可能的一些问题作了研究，主要的结论是：(1) 当分块数 N 很大时，矩阵的生成和转置会有较大的计算开销；(2) 当冗余块粒度较小时，每个块包含的生成信息将占较大比重，导致较大的带宽开销；(3) 网络编码过程中，中间节点需要等待一定数量的块到达，然后进行再编码. 如果等待的块较多，则很多中间节点的等待时间的累积将是很可观的，导致大的端到端延时；如果等待的块较少，则再编码后生成的块的线性相关的概率会增加. 在文献[39]中，作者讨论了编码中一些参数的选取.

表 2 列出了以上 3 种编码的一个比较.

表 2　多描述编码、冗余编码、网络编码的比较

	多描述编码	冗余编码	网络编码
增强网络容错性	很明显	明显	明显
提高数据差异性	不太明显	明显	很明显
满足低下载速率用户	满足	不满足	不满足
算法基础	视频编码理论等	有限域、图论	有限域、图论
在线/离线编码	一般离线编码	离线编码	在线编码
引入的带宽开销	一般较大	一般较小	一般较小

2.3　传输问题的总结

VoD/P2P 的传输问题关注于如何充分利用节点的带宽. 首先，为了保证网络中的节点都可能提供资源，多源结构成为一个首选；其次，为了适应多源结构的特点以及降低用户带宽要求，视频进行了分割；最后，为了减少数据的重复以及协商开销，并达到网络最大流，网络编码可能是一个主流.

与传输相关的另一个问题是数据驱动方式，基本的有两种：(1) 拉策略(pull-based)，下游节点根据自己的需要从上游节点请求数据. 一般包括一个协

商的过程,在协商过程中可能不能充分利用带宽,并且有延时;(2)推策略(push-based),上游节点主动将数据发送给下游节点.这种策略可以避免拉策略的缺点,但可能造成数据重复,特别是在多源结构中.虽然在采用编码后,数据重复的概率降低很多,但在网络波动过程中,由于通信不及时而造成的数据重复是不能避免的.从目前看来,一种合适的做法[40]是,每隔一段时间做 pull-based,而在这段时间过程中采用 push-based.

VoD/P2P 的传输问题,大约从 2000 年开始,一直成为该领域最热的问题.但是,大部分的研究都假设直播环境,或者热门节目的点播(类似于直播).对于一个大规模、大容量的 VoD/P2P 系统,这些假设可能并不全面.如何处理冷门与热门节目并存的系统,并在系统资源不够时作怎样的调度和折中,可能是将来的一个研究点[41].另一方面,大多数研究者都针对某一个频道的性能作优化,并且明确或不明确地假设一个节点只参加一个频道.但是在实际情况下,一个节点参加多个频道的情况是普遍的,特别是系统中的那些"准 server"节点.如何对这些节点进行调度,以让它们对系统的贡献达到最大,也是将来的潜在研究点[42].

3 存储问题

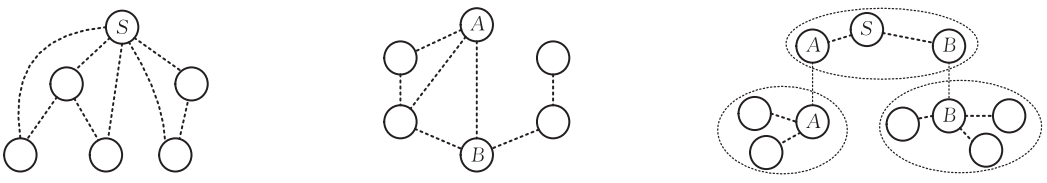
VoD/P2P 的存储希望解决如下的问题:如何将海量的视频数据部署到系统中大量的节点中,使之在动态的、异构的网络中:(1)保证数据可用性.即任意节点在任意时刻任意网络位置,都可以访问已存在于系统中的任意视频;(2)节点负载均衡.即存储应足够分散,从而避免某些节点承载大量服务而某些节点闲置的情况;(3)高效的资源定位.即对任意视频,可以迅速得到其所存储的网络位置;(4)视频的细粒度随机访问和低延迟.其中最后一点,是 VoD/P2P 存储与文件共享系统的最主要差别.本节接下来将介绍存储问题的 3 个主要方面,即资源查找结构、编码方案、存储调度.

3.1 存储中的资源查找结构

面对 P2P 网络中大量的节点,设计一个高效的资源查找算法非常重要.但在实际中,资源查找算法与节点的组织方式密切相关,因此资源查找算法的设计等价于资源查找结构的设计,并通常考虑如下因素:快速的资源定位与访问、系统的可扩展性以及小的开销.目前,常见的资源查找结构包括中心化的 P2P 网络、非中心的 P2P 网络以及层次化 Cluster 结构(图 4).

在中心化的 P2P 网络中[6,9,12],一般存在一个索引服务器(Index Server 或 Tracker),管理网络中的节点与资源.虽然这种结构有可扩展性差、单点失败等缺点,但却是实际系统中最常见的,原因如下:(1)系统实现简单;(2)索引服务器掌握整个系统的信息,因此对资源的查找非常迅速,并且容易对系统作优化;(3)P2P 网络的安全问题并没有很好解决,采用中心化的服务器对系统有更好的可控性;(4)虽然可能单点失败,但在大多数情况下,采用专门的服务器会比纯粹的 P2P 网络更稳定;(5)可扩展性问题可以通过部署多个服务器的方式解决.

与中心化 P2P 网络相对应的是非中心的 P2P 网络[13,20,43].在这种网络中路由是一个有意思的问题,即如何将消息快速发送到某个给定 ID 的节点.通常这些系统都使用了基于超立方体的路由思想.如果将网络中的节点映射成超立方体的顶点,那是 Chord[44]、Pastry[45]、Tapestry[46]的思想;而如果将超立方体的每条边进一步划分以形成更多的点(相当于笛卡儿坐标系中的点),则是 CAN[47]的思想.限于篇幅,本文对具体的路由过程不再展开.假设这样的路由算法已经实现,则可以在它的基础上构建分布式哈希(Distributed Hash Table,DHT),以实现资源定位.DHT 的基本思想是:系统中,节点与文件有相同的 ID 格式,一个文件的元信息(例如存储该文件的节点 IP 地址、文件热度等,但不包括文件本身),由具有与该文件相同 ID 的节点维护;当需要



(a)中心化的P2P网络.网络中每个节点都会直接或间接地与S通信

(b)非中心的P2P网络.通过DHT等协议,网络中的节点可以在没有S节点的帮助下发现对方

(c)层次化Cluster.A、B节点分别是下层两个Cluster的首领节点,它们与S又形成新的Cluster

图 4 存储中的资源查找结构(图中 S 表示服务器)

定位某一文件时，只要向那个与文件有相同 ID 的节点发送请求即可. 当然，实际的实现会复杂一些. 当系统中节点数为 n 时，DHT 一般可以在 $O(\log n)$ 或者 $O(dn^{1/d})$（d 是维度常数）步内定位资源. 由于 DHT 将负载分散到网络中，可以达到很好的可扩展性；由于系统没有中心，也避免了单点失败. 但是 DHT 的查询时间通常比基于中心服务器的结构要长，当网络不稳定时维护开销很大，并且存在安全问题，系统可控性差. 另一方面，DHT 是基于 ID 定位资源的，所以一般不考虑网络位置（Pastry[45] 等除外），也不支持关键字的查询.

层次化 Cluster 的思想来源于无线传感器网络：按一定的规则将若干个节点组成 Cluster，每个 Cluster 选出首领，这些首领组成新的 Cluster，从而形成一个层次化的结构. 在 VoD/P2P 网络中，节点可以根据网络位置[6-7,48-49]，相近的节点组成一个 Cluster；也可以将内容相似[50] 的节点组成一个 Cluster. 当一个节点需要定位资源时，可以从自己所处的 Cluster 开始，逐层往上查询. 在这类系统中，为了提高系统的性能，一般将最上层的首领节点设置为 server. 由于 P2P 网络的动态性，实现一个自适应的层次化 Cluster 结构比较复杂.

3.2　存储中的编码

在 VoD/P2P 系统中，存储编码主要是为了提高数据的可用性，常用的方案是冗余编码. 事实上，编码方案已经不再是一个问题，因为它在很多容灾系统中就有了深入研究. 但是 VoD/P2P 的存储编码有其特殊性，表现在：(1) 为了减小计算开销，存储编码需要与传输编码保持一致. 存储状态下的网络编码可以看成是一种特殊的冗余编码. (2) 细粒度的编码. 例如，当用户请求 23 分钟 55 秒的数据时，编码粒度为 1 分钟的方案需要将整个第 24 分钟的数据解码出来，而编码粒度为 1 秒的方案可以只解码出那 1 秒.

图 3 给出了不同的冗余编码与数据可用性的关系. 从中可以看出，扩展的冗余块数越多，数据可用性越高，但是安全问题[39] 也是需要考虑的. 因为当冗余块较少时，可以对每一块作数字签名，以保证数据的完整性；但当冗余块很多时，数字签名变得不可行. 对于网络编码，这个问题可能更突出.

3.3　存储调度

VoD/P2P 中的存储调度用于应对将来的数据请求，它关注于如何以较小的开销（带宽、存储）来达到任意视频（冷门与热门节目）、任意时刻、任意网络位置的数据可用性. 这一小节主要介绍 3 部分内容：前缀缓存、预取技术、替换策略.

前缀缓存是指在系统中维持视频的开始部分一个较高的冗余度[51-52]. 在很多情况下，VoD 系统中的用户并不会将整段视频看完，而常常只播放视频的开头部分. 相对于视频的其它部分，视频前缀有更高的访问率，因此缓存前缀可以使存储的利用更合理. 但这只是前缀缓存提出的一个原因，而另一个更重要的因素是：由于大多数用户是从头开始播放视频的（在实际情况中，视频前缀可能是播放必须的，因为它包含了视频元数据），如果视频的前缀很容易找到并下载，将减少用户的等待时间. 在实现中，这个策略需要考虑前缀冗余度、前缀长度等参数的设置. COPACC[52] 是一个多代理多 peer 的系统（代理可看成是 server），其中视频的前缀由代理存储.

预取是指节点在带宽剩余的情况下，预先下载暂时还不会被使用的数据，这些数据将来可能会被本节点或其它节点使用. 在 VoD/P2P 系统中，预取技术主要是为了提高系统整体性能. 预取可以分为在线预取和离线预取. 前者是指节点在视频播放过程中，预先取得该视频的其它数据[53-56]；后者则是指节点在没有视频播放的时候，预先部署某个视频的数据[57]. 离线预取更多地表现为替换策略，所以这里的预取只针对在线预取. 常用的预取策略有：(1) 顺序预取，指按照播放顺序预取数据；(2) 随机预取，指随机选择要预取的数据；(3) 最少块预取，指优先预取局部或全局最稀少的块（rarest-first 策略）. 顺序预取一般会导致节点间数据差异性变小而共享困难，但另一方面，由于视频是顺序播放的，完全的随机预取或最少块预取将导致很差的播放效果. 在实际过程中，一般将一个视频分成若干段，在段内采用随机或最少块预取，而段间顺序预取. 最少块预取比随机预取效果更好，但由于需要知道更多信息，也会带来更多开销. 在 RedCarpet[55] 中，采用第 1 个段 90% 的概率随机预取，第 2 个段 10% 的概率最少块预取策略. 在 P2Proxy[54] 中，server 通过自己掌握的全局信息来引导 peer 预取数据. 在文献[56]中，He 等提出了一个有意思的预取策略. 作者试图挖掘视频内部的相关性，预测用户的 VCR 操作可能跳到的位置，以提前预取相应的数据.

节点的存储空间是有限的，替换策略决定哪些数据需要被存储，哪些数据需要被替换，以达到最佳的系统性能[43,58]. 常用的替换依据包括热度、使用率等. 热度反映了一个节目被点播的次数，通常热门

的节目需要更高的冗余度;使用率反映了一个数据块被访问的次数,使用率低的块一般优先被替换出去.但是一个好的替换策略需要考虑更多因素:(1)如何计算热度与使用率.使用全局的信息需要更多开销,而使用局部信息可能不会很准确;(2)如何适应节点的异构性.在 P2P 网络中,节点的在线时间、带宽存储性能、贡献积极性会有很大差别.如果对这些节点使用相同的替换策略可能会有问题;(3)系统在保证热门节目的同时,也需要考虑冷门节目的可用性.很多系统都研究了 peer 与 server 的协同缓存问题.在 PROP[43] 中,提出了一个缓存策略,由 proxy 缓存热门的节目,而由 peer 依据"利用度"缓存其它的节目."利用度"是通过热度变换而来,一般热门或冷门的节目有一个小的"利用度",而热度适中的节目有一个大的"利用度".

3.4 存储问题的总结

VoD/P2P 的存储问题关注于如何充分利用节点的存储.在编码方案上,冗余编码可提高数据的可用性,可能是将来的主流.在资源查找结构上,虽然 DHT 等非中心化结构有极好的可扩展性,但由于其不稳定、可控性差等,目前并非实际系统的首选;反而是层次化 Cluster 结构可以充分考虑网络位置,可能是将来的一个选择.但由于简单有效,目前实用的索引系统主要还是基于中心化结构的.

存储与传输是密不可分的,两者共用一套编码方案.通过存储调度,节点间有更多的数据共享机会,从而可以更好地利用带宽.与 P2P 直播相比,点播在实际中遇到的一个很大的问题是"无源可用",因此存储可能是比传输更重要的问题.在过去的近十年中,由于视频直播是一个主要的研究焦点,存储问题并没有引起足够的重视.虽然 VoD/P2P 可以继承文件共享系统以及基于 C/S 的 VoD 的相关成果,但这个问题的研究还远远不够.如何将存储与传输更好地结合,如何构建一个自适应的位置相关、内容相关、兴趣相关的小世界存储网络,如何设计有效的替换策略等等,都是需要进一步研究的.

4 激励问题

有关激励问题的研究最初出现在 P2P 的文件共享系统.一份对 Gnutella 的研究[59] 指出,系统中 70%的用户不共享任何文件,近 50%的请求是由 1%的用户处理的.用户这种只享受服务而不提供服务的行为,通常被称为 free-riding.在 VoD/P2P 中,普遍的 free-riding 行为将导致频繁的点播缓冲,这将迫使运营商不得不部署更多的服务器,导致系统成本上升.因此,设计一个有效的激励机制,减少或避免用户的 free-riding 行为,对 VoD/P2P 非常重要.目前,常见的激励机制可分为 3 类:Tit-for-Tat(TFT)、不对称 TFT、虚拟货币机制.

4.1 Tit-for-Tat

TFT 可以看成是博弈论的一个实例,其核心思想是:提供多少服务,获取多少服务.统计服务的粒度可以是字节数、块数、流数、上传速率等,并且可能是一个模糊的统计.这种机制可应用于一次会话中,以避免 free-riding[26-27,60];也可以基于长期行为,如积分机制、信誉机制[61-62].在 Orchard[26] 中,视频采用 MDC 编码分成多个流,一个节点下载一个流就需要上传一个流,即与其它节点进行流交换.由于流越多视频越清晰,所以有能力的节点就会倾向于上传更多的流.

虽然 TFT 可以很好地应用于 BT 等文件共享系统以避免 free-riding,但对 VoD 并不十分适用,这是因为:(1)TFT 一般通过限制下载速率来惩罚那些提供服务较少的节点,这对文件下载的影响是更长的下载时间,但对 VoD 的影响是视频不能播放,导致用户流失;(2)TFT 在文件下载中取得成功的一个重要原因是"rarest-first"的数据下载策略,但 VoD 通常要求顺序地下载与播放,因此刚开始播放的节点由于缺乏其它节点感兴趣的数据而不能提供服务,成为 TFT 的受害者.

TFT 是一种公平的机制,但这也导致了它的缺点:那些服务能力差的节点将得不到及时的服务,可能无法生存.

4.2 不对称 TFT

不对称 TFT 是针对 TFT 的缺点提出的,它要求服务能力强的节点贡献更多资源,以帮助服务能力差的节点.在 taxation[63] 中,不同的节点有不同的上传指标,这被抽象成税收,服务能力强的节点的税率更大一些.系统将这些收取的税当成基本社会福利,平分给系统中的节点.也就是说,节点即使不提供任何服务,也可以拥有"基本社会福利"的服务.不对称 TFT 的一个极端情况是:所有加入到系统中的节点都要求毫不保留地提供服务.这实际上是目前很多商业系统的做法[64],它们通过客户端程序强制用户贡献资源,而对资源贡献的多少不加区分,节点不会因为贡献资源多而得到更好服务,也不会因为贡献资源少而得到较差服务.

与 TFT 不同，不对称 TFT 更多地追求平均化，因此从某种意义上说它不能算是激励机制. 但这种机制有如下的优点：(1) 通过照顾服务能力差的节点扩大了用户面，这对以广告作为重要收入的商业 VoD/P2P 系统非常重要；(2) 强制性地利用系统中所有节点，特别是服务能力强的节点的资源，提高了系统的整体性能；(3) 在不追求公平的条件下，这种机制实现比较简单.

不对称 TFT 对服务能力强的节点是不公平的，容易造成用户反感，这些节点在点播结束后倾向于立即退出系统，从而影响系统性能.

4.3　虚拟货币机制

虚拟货币机制[65-67]的思想是：节点提供服务增加虚拟货币，获取服务减少虚拟货币. 与上文提到的两类机制的差别是，虚拟货币机制是开放性的，它不局限于系统内部，因为虚拟货币可以与现实货币相互转换(出于安全的考虑，一般不允许用虚拟货币兑换现实货币). 也就是说，提供服务并不是增加虚拟货币的唯一方式，节点可以通过其它途径增加自己的虚拟货币，如向系统购买. 因此，即使是服务能力差的节点，只要拥有足够的虚拟货币，同样可以取得好的服务.

虚拟货币机制通常要求较高的安全性，以抵制货币造假等攻击，它一般涉及认证、加密、签名等操作，因此通常依赖于 PKI. 由于保证绝对安全需要很大的开销，实际系统只要求足够安全，即做到攻击是可发现的、可跟踪的、无获益的. 在 Karma[67] 中，系统通过签名为每个用户“打造”一些货币，当用户 A 转让自己的货币给用户 B 时需附上自己的签名，同样 B 转让给用户 C 时也要附加签名，由此货币本身累积了交易历史，当历史积累到一定程度时需要重新更新货币，并删除历史.

虚拟货币机制是公平的，并且照顾到了服务能力差的节点，但是在实现中，除了安全性以外，有以下的问题是需要考虑的：(1) 服务能力差的用户需要向系统购买虚拟货币，即系统中增加了收费用户. 面对目前国内互联网服务大多免费的现状，向用户收费存在一定的难度，操作不当可能导致用户流失；(2) 服务能力强的节点可能累积大量的虚拟货币，系统需要提供一条途径兑换这些货币，否则这些虚拟货币将与积分一样意义不大，从而打击这些节点进一步提供服务的积极性.

4.4　激励问题的总结

激励问题关注于如何刺激系统中的节点贡献资源，以提高系统的整体性能. 在设计激励机制时，即要考虑公平性，又要考虑节点服务能力的差异性. TFT 是一种公平的机制，但没有照顾好服务能力弱的节点；不对称 TFT 考虑了所有节点的服务能力，但并不公平；虚拟货币机制兼顾了两方面的因素，但存在着安全性与实际推广的挑战.

激励有两个目标：(1) 刺激节点提供更多的服务；(2) 刺激节点在线更长的时间. 到目前为止，绝大多数的激励机制都专注于第 1 个目标，通过客户端程序的控制，这个目标的实现也较简单. 相比之下，第 2 个目标的实现较困难. 很多 VoD/P2P 用户通常在完成观看后及时地终止客户端，因为继续保持在线并没有多少好处，且可能影响其它的应用. 虚拟货币机制是有可能实现第 2 个目标的，一个理想的情况是，P2P 用户在使用服务后仍乐意保持在线，因为提供服务将获得虚拟货币，而这些虚拟货币可以通过某些途径转化为其它的利益. 由于这部分用户只提供服务而不消耗服务，将极大地提升整个系统的性能，从而降低 VoD/P2P 系统的部署和运行成本.

5　总　结

本文介绍了 VoD/P2P 系统的研究历史、研究现状以及面临的挑战. 由于 VoD 的高带宽要求、P2P 网络的不稳定性和异构性，有限的资源可用性通常是这类系统的一个主要挑战. 因此，本文关注于如何充分利用系统中有限的资源，在 3 个方面进行了综述：(1) 数据传输. 讨论如何充分利用节点的带宽资源；(2) 数据存储. 讨论如何充分利用节点的存储资源；(3) 激励机制. 讨论如何刺激节点贡献更多资源.

经过近十年的发展，VoD/P2P 系统的一个新的发展趋势是：虽然 P2P 网络是动态的、异构的，但其统计特性是好的，如果一个服务依赖于很多的节点，就不会因为个别节点的失效而带来很大影响. 这个思想具体表现在：(1) 在传输中，通过多源结构、编码技术，让很多个节点同时为一个节点服务；(2) 在存储中，通过编码，将一个视频分布到多个节点中.

在 VoD/P2P 的设计过程中，还有诸如安全、数字版权等一系列问题. 本文由于篇幅，并没有对它们进行讨论，但构建一个实用的 VoD/P2P 系统，所有这些问题都是需要考虑的.

参 考 文 献

[1] Dan A, Sitaram D, Shahabuddin P. Scheduling policies for an on-demand video server with batching//Proceedings of the 2nd ACM Multimedia Conference. San Francisco, 1994: 15-23

[2] Aggarwal C C, Wolf J L, Yu P S. On optimal batching policies for video-on-demand storage servers//Proceedings of the 3rd IEEE International Conference on Multimedia Systems. Japan, 1996: 253-258

[3] Hua K A, Cai Y, Sheu S. Patching: A multicast technique for true video-on-demand services//Proceedings of the 6th ACM International Conference on Multimedia. Bristol, 1998: 191-200

[4] Jannotti J, Gifford D K, Johnson K L, Kaashoek M F, OToole J. Overcast: Reliable multicasting with an overlay network//Proceedings of the 4th Symposium on Operating System Design and Implementation. San Diego, 2000: 14-14

[5] Deshpande H, Bawa M, Garcia-Molina H. Streaming live media over peer-to-peer network. Stanford University, Technical Report CS-TR-01-501, 2001

[6] Banerjee S, Bhattacharjee B, Kommareddy C. Scalable application layer multicast//Proceedings of the ACM SIGCOMM. Pittsburgh, 2002, 32(4): 205-217

[7] Tran D A, Hua K A, Do T. ZIGZAG: An efficient peer-to-peer scheme for media streaming//Proceedings of the IEEE INFOCOM. San Francisco, 2003: 1283-1292

[8] Do T T, Hua K A, Tantaoui M A. P2VoD: Providing fault tolerant video-on-demand streaming in peer-to-peer environment//Proceedings of the IEEE International Conference on Communications. Paris, 2004: 1467-1472

[9] Guo Y, Suh K, Kurose J, Towsley D. P2Cast: Peer-to-peer patching for video on demand service. Multimedia Tools and Applications, 2007, 33(2): 109-129

[10] Zhang L, Lo K T. A peer-to-peer architecture for on-demand video streaming on Internet//Proceedings of the International Conference on Communications, Circuits and Systems. Chengdu, 2004: 525-528

[11] Kusmierek E, Dong Y, Du D H C. Loopback: Exploiting collaborative caches for large-scale streaming. IEEE Transactions on Multimedia, 2006, 8(2): 233-242

[12] Padmanabhan V N, Wang H J, Chou P A, Sripanidkulchai K. Distributing streaming media content using cooperative networking//Proceedings of the 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video. Miami, 2002: 177-186

[13] Castro M, Druschel P, Kermarrec A-M, Nandi A, Rowstron A, Singh A. SplitStream: High-bandwidth multicast in cooperative environments//Proceedings of the 19th ACM Symposium on Operating Systems Principles. NY, 2003: 298-313

[14] Jiang X, Dong Y, Xu D, Bhargava B. GnuStream: A P2P media streaming system prototype//Proceedings of the International Conference on Multimedia and Expo. Baltimore, 2003: 325-328

[15] Kostic D, Rodriguez A, Albrecht J, Vahdat A. Bullet: High bandwidth data dissemination using an overlay mesh. ACM SIGOPS Operating Systems Review, 2003, 37(5): 282-297

[16] Hefeeda M, Habib A, Botev B, Xu D, Bhargava B. PROMISE: Peer-to-peer media streaming using CollectCast//Proceedings of the ACM Multimedia. Berkeley, 2003: 45-54

[17] Magharei N, Rejaie R. PRIME: Peer-to-peer receiver-driven mesh-based streaming//Proceedings of the 26th IEEE International Conference on Computer Communications. Anchorage, 2007: 1415-1423

[18] Zhang X, Liu J, Li B, Yum Y-S P. CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming//Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Miami, 2005: 2102-2111

[19] Liao X, Jin H, Liu Y, Ni L M, Deng D. AnySee: Peer-to-peer live streaming//Proceedings of 25th IEEE International Conference on Computer Communications. Barcelona, 2006: 1-10

[20] Yiu W-P K, Xing J, Chan S-H G. VMesh: Distributed segment storage for peer-to-peer interactive video streaming. IEEE Journal on Selected Areas in Communications, 2007, 25(9): 1717-1731

[21] Siddhartha A, PSaikat G, PChristos G, Dinan G, Pablo R. Is high-quality VoD feasible using P2P swarming? //Proceedings of the 16th International Conference on World Wide Web. Banff, 2007: 903-912

[22] Purandare D, Guha R. An alliance based peering scheme for peer-to-peer live media streaming//Proceedings of Workshop on Peer-to-Peer Streaming and IP-TV. Kyoto, 2007: 340-345

[23] Zhou M, Liu J. A hybrid overlay network for video-on-demand//Proceedings of the IEEE International Conference on Communications. Seoul, 2005: 1309-1313

[24] Wang F, Xiong Y, Liu J. mTreebone: A hybrid tree/mesh overlay for application-layer live video multicast//Proceedings of the 27th International Conference on Distributed Computing Systems. Toronto, 2007: 49-49

[25] Lu M T, Wu J C, Peng K J, Huang P, Yao J J, Chen H H. Design and evaluation of a P2P IPTV system for heterogeneous networks. IEEE Transactions on Multimedia, 2007, 9(8): 1568-1579

[26] Mol J D, Epema D H P, Sips H J. The orchard algorithm: Building multicast trees for P2P video multicasting without free-riding. IEEE Transactions on Multimedia, 2007, 9(8): 1593-1604

[27] Pouwelse J A, Taal J R, Lagendijk R L, Epema D H J, Sips H J. Real-time video delivery using peer-to-peer bartering

networks and multiple description coding//Proceedings of the IEEE International Conference on Systems, Man and Cybernetics. Hague, 2004: 4599-4605

[28] Fitzek F H P, Can B, Prasad R, Katz M. Overhead and quality measurements for multiple description coding for video services//Proceedings of the International Symposium on Wireless Personal Multimedia Communication. Italy, 2004: 524-428

[29] Luby M, Mitzenmacher M, Shokrollahi A, Spielman D. Efficient erasure correcting codes. IEEE Transactions on Information Theory, 2001, 47(2): 569-584

[30] Luby M. LT codes//Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science. Vancouver, 2002: 271-280

[31] Shokrollahi A. Raptor codes. IEEE Transactions on Information Theory, 2006, 52(6): 2551-2567

[32] Nguyen T, Zakhor A. Distributed video streaming with forward error correction//Proceedings of the 12th International Packet Video Workshop. Pittsburgh, 2002: 212-223

[33] Ahlswede R, Ning C, Li S Y R, Yeung R W. Network information flow. IEEE Transactions on Information Theory, 2000, 46(4): 1204-1216

[34] Wu C, Li B. rStream: Resilient and optimal peer-to-peer streaming with rateless codes. IEEE Transactions on Parallel and Distributed Systems, 2008, 19(1): 77-92

[35] Gkantsidis C, Rodriguez P R. Network coding for large scale content distribution//Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Miami, 2005: 2235-2245

[36] Wang M, Li B. Network coding in live peer-to-peer streaming. IEEE Transactions on Multimedia, 2007, 9(8): 1554-1567

[37] Thomos N, Frossard P. Raptor network video coding//Proceedings of the International Workshop on Mobile Video. Augsburg, 2007: 19-24

[38] Wang M, Li B. How practical is network coding? //Proceedings of the 14th IEEE International Workshop on Quality of Service. New Haven, 2006: 274-278

[39] Li J. Erasure resilient codes in peer-to-peer storage cloud//Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. Toulouse, 2006: IV-IV

[40] Tang Y, Luo J, Zhang M, Yang S, Zhang Q. Deploying P2P networks for large-scale live video-streaming service. IEEE Communications Magazine, 2007, 45(6): 100-106

[41] Wu C, Li B, Zhao S. Multi-channel live P2P streaming: Refocusing on servers//Proceedings of the 27th Conference on Computer Communications. Phoenix, 2008: 1355-1363

[42] Wang F, Liu J, Xiong Y. Stable peers: Existence, importance, and application in peer-to-peer live video streaming//Proceedings of the 27th Conference on Computer Communications. Phoenix, 2008: 1364-1372

[43] Guo L, Chen S, Zhang X. Design and evaluation of a scalable and reliable P2P assisted proxy for on-demand streaming media delivery. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(5): 669-682

[44] Stoica I, Morris R, Karger D, Kaashoek M F, Balakrishnan H. Chord: A scalable peer-to-peer lookup service for internet applications//Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. San Diego, 2001: 149-160

[45] Rowstron A, Druschel P. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems//Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms. Heidelberg, 2001: 329-350

[46] Zhao B Y, Kubiatowicz J D, Joseph A D. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. UCB: Technical Report CSD-01-114, 2001

[47] Ratnasamy Sylvia, Francis P, Handley M, Karp R, Shenker S. A scalable content-addressable network//Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. San Diego, 2001: 161-172

[48] Hefeeda M M, Bhargava B K, Yau D K Y. A hybrid architecture for cost-effective on-demand media streaming. Computer Networks: The International Journal of Computer and Telecommunications Networking, 2004, 44(3): 353-382

[49] Wan K H, Loeser C. An overlay network architecture for data placement strategies in a P2P streaming network//Proceedings of the 18th International Conference on Advanced Information Networking and Applications. Fukuoka, 2004: 119-125

[50] He X, Tang X, You J, Xue G. Network-aware multicasting for video-on-demand services. IEEE Transactions on Consumer Electronics. 2004, 50(3): 864-869

[51] Chan C, Huang S, Chen H, Tung W, Wang J. An application-level multicast framework for large scale VOD services //Proceedings of the 11th International Conference on Parallel and Distributed Systems. Fuduoka, 2005: 98-104

[52] Alan T S, Liu J, Lui J C S. COPACC: An architecture of cooperative proxy-client caching system for on-demand media streaming. IEEE Transactions on Parallel and Distributed Systems, 2007, 18(1): 70-83

[53] Sharma A, Bestavros A, Matta I. dPAM: A distributed prefetching protocol for scalable asynchronous multicast in P2P systems//Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Miami, 2005: 1139-1150

[54] Lee G J, Choi C K, Choi C Y, Choi H K. P2Proxy: Peer-to-peer proxy caching scheme for VOD service//Proceedings of the 6th International Conference on Computational Intelligence and Multimedia Applications. Las Vegas, 2005: 272-277

[55] Annapureddy S, Gkantsidis C, Rodriguez P. Providing video-on-demand using peer-to-peer networks//Proceedings of the Internet Protocol Television Workshop. Edinburgh, 2006: 1-14

[56] He Y, Liu Y. Supporting VCR in peer-to-peer video-on-demand//Proceedings of the IEEE International Conference on Network Protocols. Beijing, 2007: 328-329

[57] Kyoungwon S, Diot C, Kurose J, Massoulie L, Neumann C, Towsley D, Varvello M. Push-to-peer video-on-demand system: Design and evaluation. IEEE Journal on Selected Areas in Communications, 2007, 25(9): 1706-1716

[58] Guo P, Yang Y, Guo H. Cooperative caching for peer-assisted video distribution//Proceedings of the 13th International Multimedia Modeling Conference. Singapore, 2007: 135-144

[59] Adar E, Huberman B A. Free riding on Gnutella. First Monday, 2000, 5(10): 1-10

[60] Pianese F, Perino D, Keller J, Biersack E W. PULSE: An adaptive, incentive-based, unstructured P2P live streaming system. IEEE Transactions on Multimedia, 2007, 9(8): 1645-1660

[61] Kung H T, Wu C H. Differentiated admission for peer-to-peer systems: Incentivizing peers to contribute their resource//Proceedings of the 1st Workshop on Economics of Peer-to-Peer systems. Berkeley, 2003: 1-6

[62] Shi N, Dai Q. A novel incentive mechanism improving peer-to-peer on-demand streaming//Proceedings of the International Conference on Communications, Circuits and Systems. Guilin, 2006: 91-95

[63] Chu Y H, Chuang J, Zhang H. A case for taxation in peer-to-peer streaming broadcast//Proceedings of the ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems. Portland, 2004: 205-212

[64] Huang Y, Fu T Z J, Chiu D M et al. Challenges, design and analysis of a large-scale P2P-vod system. ACM SIGCOMM Computer Communication Review, 2008, 38(4): 375-388

[65] Golle P, Kevin L B, Mironov I. Incentives for sharing in peer-to-peer networks//Proceedings of the ACM Conference on Electronic Commerce. Tampa, 2001: 264-267

[66] Antoniadis P, Courcoubetis C. Market models for P2P content distribution//Proceedings of International Workshop on Agents and Peer-To-Peer Computing. Bologna, 2002: 138-143

[67] Garcia F D, Hoepman J H. Off-Line Karma: A decentralized currency for static peer-to-peer and grid networks//Proceedings of the 5th International Networking Conference. NY, 2005: 325-332

SHEN Shi-Jun, born in 1982, Ph. D. candidate. His research interests include peer-to-peer networks, streaming media and mobile computing in wireless ad hoc networks.

LI San-Li, born in 1935, professor, Ph. D. supervisor, member of the Chinese Academy of Engineering (CAE). His research interests include high performance computing, grid computing, mobile computing and media streaming.

Background

After the great success of Napster in 1999, the era of peer-to-peer (P2P) applications has arrived. In the past decade, hundreds of P2P applications, such as Gnutella, Bittorrent, eDonkey, PPStream, PPLive, UUSee, and Joost, have witnessed tremendous growth among end users all around the world. Today, more than 60% of Internet backbone traffic is P2P (data from CacheLogic). Not only is P2P technology popular for file-sharing systems, but it is also powerful for live video streaming applications.

P2P based Video-on-Demand (VoD/P2P) as a more attractive application, has recently received a tremendous amount of attention. There is a strong belief among telecommunication companies that this market will expand exponentially in the next few years. However, there are many challenges in designing, implementing, and deploying such a system. This paper outlines the components of existing VoD/P2P architectures and surveys approaches to their design. Three major aspects of VoD/P2P design are discussed, i. e. data transmission, data storage and incentive mechanisms. For each aspect, the authors identify issues that are yet to be addressed or have not received sufficient attentions.

This work is supported in part by the National Excellent Courses Integration Project (JPKC-5) which sets educational video streaming on Internet as one of its goals. In the past years, the authors participated in the development of a large-scale cluster based VoD system called Nova. Currently, the authors are designing and implementing another user-oriented, peer-to-peer based VoD system.