计算机科学2002Vol. 29№. 9

保障 QoS 的 P2P 分布式 VOD 系统的设计

Design of a QoS Guaranteed Distributed P2P VOD System

刘 鹏 都志辉 李三立 陈 渝 朱子玉 黄震春 顾 雷

(清华大学计算机科学与技术系 北京100084)

Abstract To decentralize the VOD services by using the booming P2P technology, of which any user acts as both the consumer and the provider, will eliminate the bottleneck of traditional VOD systems and remove the heavy task to provide film resource. The paper provides three choices to implement the system with a comparison of each other. Policies to guarantee the QoS are discussed as well, such as to dynamically select the server with the highest QoS value and replace it when its QoS drops below a threshold, etc. The paper is the first one that presents a method to describe the volume capability of a P2P video node. By using the method, it gets proving results of the feasibility of the P2P VOD system from an elaborate experiment.

Keywords P2P, VOD, QoS, Distributed system, Broad band network

引言

网络对等计算模式 P2P(Peer-to-Peer)[1,4,5,18]是当今的研究热点之一。与较早的 C/S(Client/Server)计算模式相比，该模式没有明显的 Client 端与 Server 端的区别，可以认为它的任何一端同时具有 Client 和 Server 的功能。

事实上，早在 Internet 尚未形成的时候，大型计算机之间采用的就是对等端通信方式，从这个意义上讲，P2P 至少有四、五十年的历史了。不过，相比现在的 P2P，当时还非常原始。首先，那时能够通信的对象非常有限，不像在 Internet 环境中可以与任意一台主机通信；其次，当时通信的主要目的是传送私有格式的文件和数据，而现在更多是在开放的平台上进行资源共享、信息交换和网络计算等；再次，通信的基础设施如网络、微处理器、存储器、软件平台等在过去几十年里有了巨大的进展。例如，网络带宽和微处理器的处理能力已经连续几十年保持了每年60%的增长。

在国外，P2P 的研究热潮是由一个非常成功的 P2P 应用——Napster[6]带动起来的。Napster 是一种用户共享 MP3音乐的软件。它的"以1易 N"的逻辑受到用户的普遍欢迎，在其鼎盛时期，美国大学70～80%的网络带宽都被它占用。另外，各种聊天软件(如 ICQ，OICQ)也属于 P2P 应用，它们同样掀起了热潮。可以这样认为，这些都属于窄带时代成功的 P2P 应用，因为它们所交换的媒体(压缩音乐和文本)即使在 Modem 之间也能得到很好的传输。后来又出现了类似于 Napster 的、以文件共享为特征的 Imesh[14]，MojoNation[15]，Freenet[16]，Gnutella[17]等系列的 P2P 应用。除了文件共享外，一些其他应用研究项目也采纳了 P2P 机制，比如在医学权威杂志《Nature Medicine》上发表的一篇文章[9]介绍了牛津大学等研究单位的学者，根据大众自愿的原则，利用众多微机的大量空闲时间，用 P2P 分布式计算来解决关于癌症方面的问题；而文[10]则公布了斯坦福大学的研究者如何通过屏幕保护程序，试图通过 P2P 计算解开蛋白质结构之谜；伯克利大学的 SETI@home 项目[19]则通过 P2P 分布式计算来研究外星生命。许多学者还在网络学习、电子商务、计算网格等领域对 P2P 计算展开研究[10,12,13]，还有的学者对 P2P 的适用范围进行了探讨[3]。

国外一些著名公司展开对 P2P 支持平台的研究[2]。Microsoft 在其.NET 框架下提出了支持 P2P 的.NET My Services(以前叫做 Hailstorm)[7]，企图实现以用户为中心而不是传统的以计算机为中心的计算模式；而 SUN 则推出了 JXTA 项目[8]，试图实现一个与语言、操作系统和网络协议无关的 P2P 应用开发支持环境。

P2P 之所以在 Internet 和 C/S 模式已经非常成熟的背景下走向前台，是因为目前网络的发展水平仍然无法满足迅速膨胀的用户需求。一方面，用户的数量呈爆炸性增长；另一方面，原来以文本页面为主的 Internet 服务转向了以多媒体为主，如：VOD(Video On Demand)、音乐、图片、3D、互动游戏、远程教学、远程医疗以及互动购物等。由于这些突变，使得原来强大的集中式服务器不堪重负，服务质量 QoS(Quality of Service)得不到保障。例如，访问过宽带电影网站的人都会有这样的感受：一到高峰时段，几乎所有的宽带网站都很难连上，即使连接上了，电影的播送也是时断时续。另外，网络电影的显示画面都较小，画面质量很粗糙。之所以如此，是因为无论是集中式服务器本身，还是它的网络带宽，都构成系统的瓶颈。要想消除这个瓶颈，最好的办法是将它的服务分散化，使系统中的任意主机既享受服务，也提供服务--这就是 P2P 的策略。本文的目标就是要在宽带环境中实现一个实用的、保障 QoS 的 P2P 系统——分布式 VOD 系统。

设计方案

VOD 系统具有很大发展潜力：相比家庭 VCD/DVD 而言，它可以提供多得多的片源；相比电视和有线电视而言，用户的自主权更大，可以任意确定播放的时间和内容。但即使在宽带环境中，VOD 系统的瓶颈问题还是普遍存在，这主要集中在服务器端。由于视频流需要占用的网络带宽是音频流的数倍乃至上百倍，是文字流的成千上万倍，因此视频服务器端的网络带宽很难满足大用户量的需要。例如，假设某个 VOD 服务器拥有100M 的网络带宽，每个视频流占用300K 的带宽，那么这个服务器所能承载的最大用户数仅为341个，这比一般的 Web 网站所能承载的用户数少得太多。对一个 VOD 网站而言，这个数据根本不具有规模经济性。除了带宽问题，服务器本身的 I/O 吞吐能力也是一个严重的问题。虽然现在

刘 鹏 博士生。都志辉 博士后。李三立 中国工程院院士，博士生导师。陈 渝 博士后。朱子玉 博士生。黄震春 博士。顾 雷 博士生。主要从事网络计算、分布/并行体系结构的研究工作。

可以用集群技术的良好可扩展性来提高服务器的I/O能力，但其中每一个服务结点所能承载的视频用户数有限，而且成倍增加结点数量也会成倍增加成本。

如果VOD系统的QoS问题不能解决，就会大大影响它的普及。解决这个问题的关键就是消除VOD系统的瓶颈——利用P2P技术，将集中的VOD服务分散化，形成负载均衡、播放质量有保障的分布式VOD系统。在此，本文提出用P2P技术实现分布式VOD系统的三种不同方案，并比较各自优缺点。所有这些方案都默认这样的规则：任何加入到系统的用户，在自由享用系统中的影片资源时，也有义务向其他用户提供服务。也就是说，各用户既是资源的消费者，也是服务者。

方案1：共享用户影片资源

如图1所示。在这个系统中，服务器负责整理并综合用户的片源信息，对影片进行归类，列出所有影片的清单，提供影片的内容介绍，帮助用户选定影片，并引导用户的播放器找到影片的提供者。这个服务器不提供影片的内容，只提供影片的目录，故称为目录服务器。

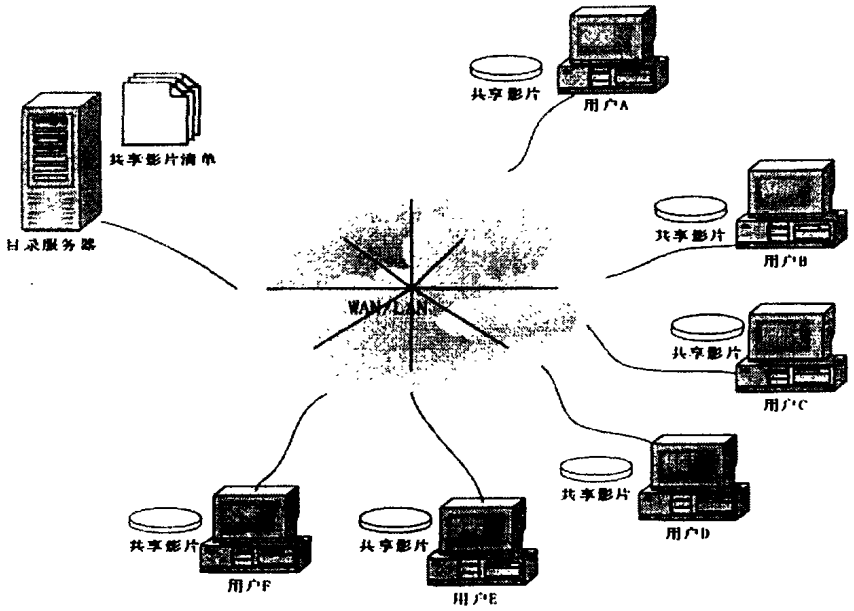

图1　共享用户影片资源

当用户A使用这个分布式VOD系统时，首先应该确保本机的共享目录中存放了一定数量的共享影片资源。这些影片的目录信息会在连接目录服务器时自动上传，由目录服务器整理收录。

由于不同用户提交的影片不可避免地会存在重复，目录服务器需要归并这些重复项，并记下每一部影片的所有提供者的地址。需要指出的是，当前能够访问的影片及相应提供者的信息总是动态变化的，目录服务器中的清单需要与系统的当前状况随时保持同步。为了方便用户选择影片，应统一影片的片名，并将其归类（如故事片、动画片等），这就需要在服务器中保存一个片名类别库。同时，为了让用户有更多的参考信息，目录服务器还可以保存一个影片资料库，提供每部影片的相关资料，如内容简介、影评、用户打分等。当用户A通过访问目录服务器确定要播放的影片之后，服务器就把提供该影片播放服务的用户清单传给用户A的播放器。

用户A的播放器向这个清单的影片提供者发出测试数据包，然后根据反馈者的时延、数据正确性、播放速度、播放的稳定性等参数选出一个播放质量最高的用户作播放服务器，清单中其他可用的服务者作为备用。整个流程如图2所示。当然，在播放的过程中，各种不稳定的状况都有可能出现，如网络拥阻、播放服务器负载增加或中断播放（断网或关机）等。为了动态地保障QoS，用户A的播放器在播放过程中，应当自动评估当前的QoS，如果不断发生QoS下降的情况，则事先探询备用服务器，当确认QoS下降到某个域值之下时，就自动切换到一台最佳的备用服务器上。这个切换过程对用户是透明的，切换时延带来的副作用只是画面的短暂停顿。如果清单中的备用服务器都不可用，则用户A的播放器向目录服务器发出请求，重新获取最新的服务者清单。只有当所有的服务提供者都已退出系统时，才中断用户A的播放。在这个过程中，用户A的播放QoS得到了最大程度的保障。

方案2：分散缓存服务器影片资源

上述共享用户影片资源的方案应该说对用户是很有吸引力的。不过，上述这种共享方式也可能存在一定问题。首先是版权问题——由于它提供了一个用户之间可以任意交换影片的平台，客观上助长了用户任意扩散盗版的行为。Napster就曾因相同的问题而被美国音乐协会起诉。从长远来看这个问题值得重视；其次是用户扩散某些非法影片的问题。当然，一部分已知的非法影片可以在目录服务器统一片名时加以排除，但未知的和被用户改过名字的非法影片却很难管理；最后是用户的规模问题——它与集中式VOD系统截然相反，其用户规模越大，QoS才越容易得到保障。当用户量较小时，共享用户影片资源的方案就暴露了其缺点：一方面，每个用户可选择的影片范围小，另一方面，如果一个用户正在观看的某部影片只有一个提供者，一旦该提供者退出系统，该用户就得中断播放。

上述前两个问题都是有关影片内容的问题。如果要严格进行影片内容的控制，最好的办法就是由服务器端提供所有的影片资源。当然，这样做之后，上述第三个问题——规模小就影响QoS的问题——也就迎刃而解了。同时，为了防止单源点造成的瓶颈效应，本方案利用P2P技术，实现分散缓存。

例如，用户 B 和用户 C 播放了某一部影片，它们的计算机里都会缓存它。这时，如果用户 A 也想观看该影片，它的播放器就会从服务器、用户 B 和用户 C 中挑出 QoS 最好的作为播放服务器，如图3所示。这样，用户 A 一般都能得到较好 QoS 的服务。当用户 A 在播放该影片的同时，也将缓存它而成为其服务提供者之一。

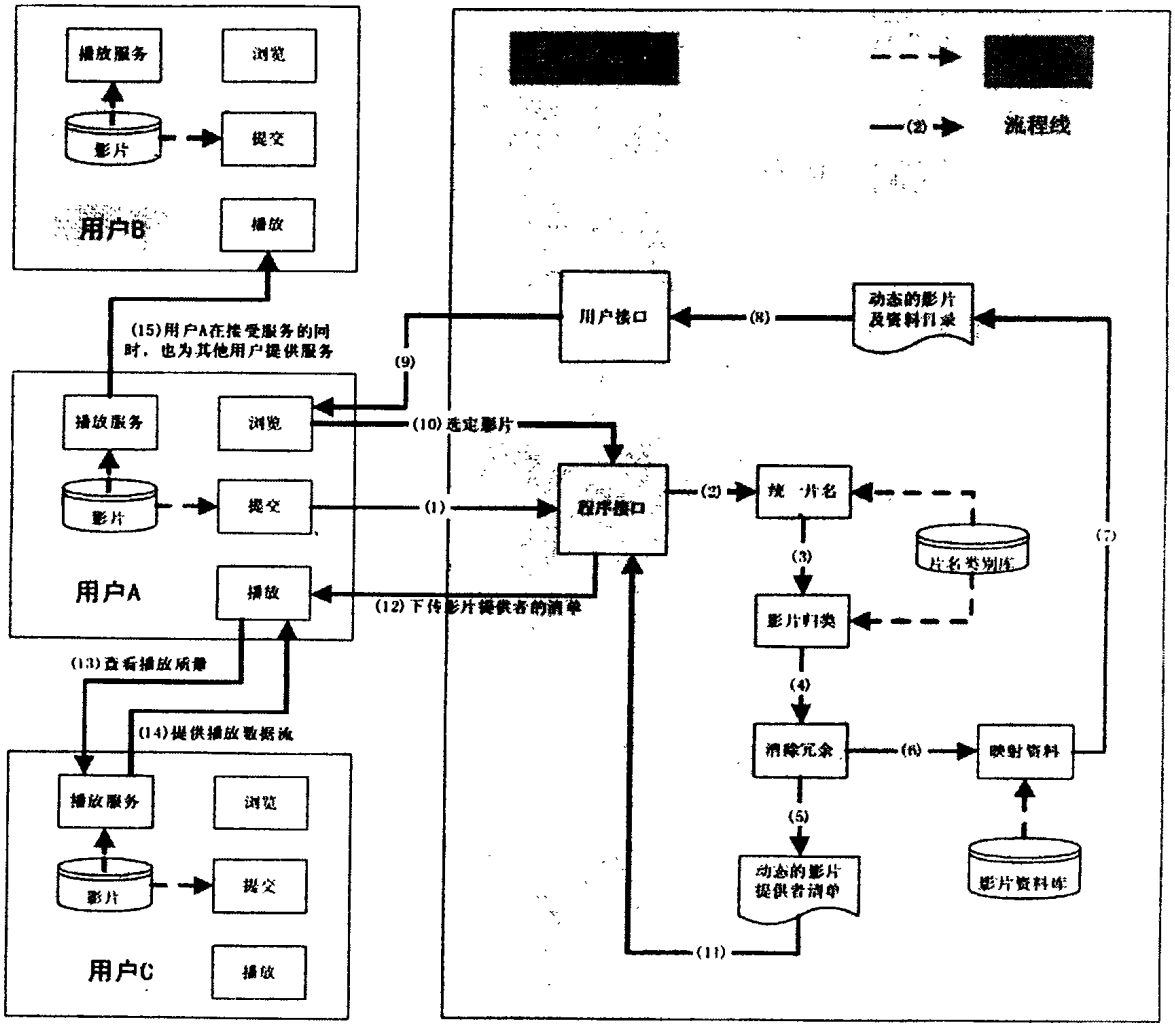

图2　用户 A 从提交影片到获得服务的处理流程

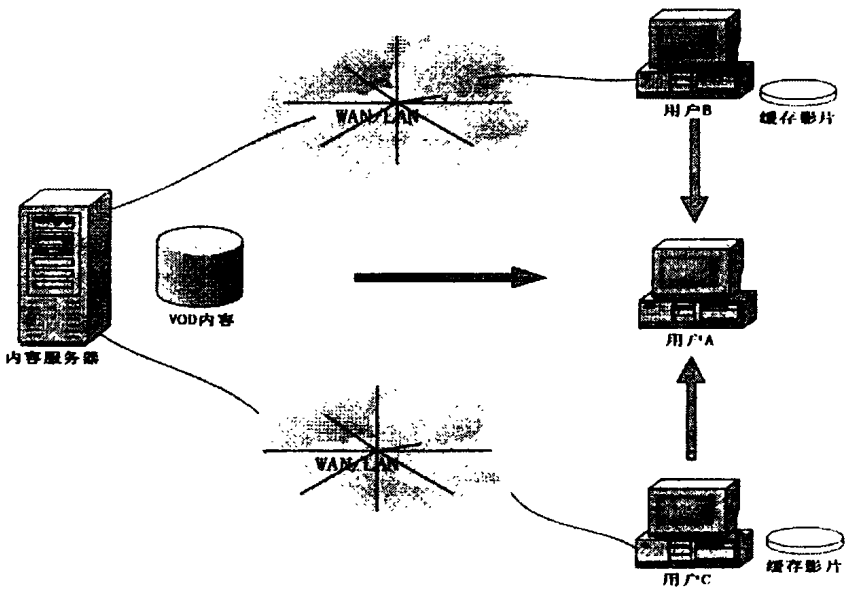

图3　选择一个 QoS 最好的服务提供者

由于用户能够缓存的影片数量是有限的，所以应当有一个淘汰算法，将不太经常被其他用户访问的影片淘汰掉。为了保障 QoS，这个方案也有当 QoS 下降到阈值时动态切换服务提供者的问题。

方案3：同时共享服务器和用户影片资源

第2个方案解决了影片的内容问题，不过，所有内容都要由服务器提供并不断更新也是一个不小的经济和管理负担。而且，过于保守的服务器也难以引起网友的强烈兴趣。为此本文提出一个实现复杂而又有更多灵活性的方案——就是服务器和用户都能提供影片的内容，而且用户还能缓存影片而使系统资源分散，如图4所示。

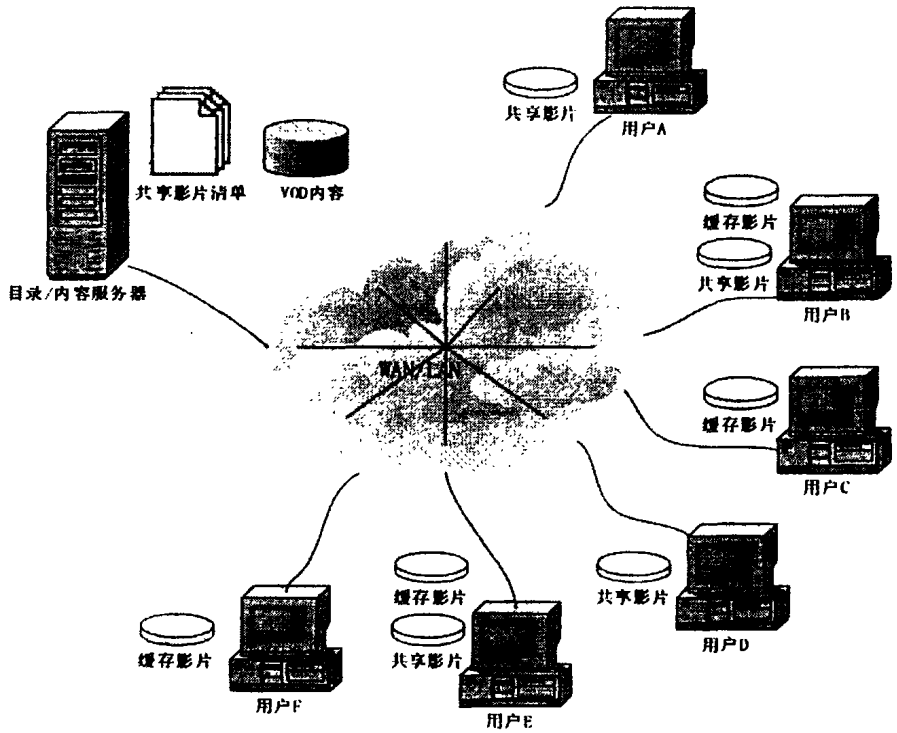

图4　服务器和部分用户都提供影片资源并分散缓存之

在这个系统中，任何用户不必提供共享影片就能加入，只不过提供了共享影片的用户在享受服务时可以拥有一定的优先权。任何用户都能缓存它所播放的影片，这些影片可能来自服务器，也可能来自其他用户。如果一个用户所提供的共享影片足够多或很受欢迎，达到一定的负载，那么该用户的计算机就不必再缓存其他的影片了。

这个系统的实现较前两个系统更复杂，但它更加灵活。它既可以按方案1的方式进行工作，又可以按方案2的方式进行工作，还可以按二者结合的方式进行工作。在版权的问题上，它解决了一部分，而且可以灵活控制这部分所占的比重。由于 P2P 是大势所趋，相信以后会有相关的法律，较好地解决网上影片资源的共享版权问题。

上述三种方案的相对优缺点总结如下表，在实现时选择何种方案需要根据实际情况综合考虑。

	优点	缺点
方案1	1. 实现简单 2. 容易调动用户的参与热情	1. 版权问题 2. 非法影片问题 3. 用户规模太小就难以保障 QoS
方案2	1. 实现简单 2. 克服了方案1的缺点	1. 过于保守 2. 需要提供所有内容，经济和管理负担重
方案3	同时具有方案1、2的优点，可以灵活调整	1. 实现复杂 2. 仍然存在一定版权和非法影片问题

性能分析

下面分析本文提出的 P2P 分布式 VOD 系统的可行性，主要解决这样一个问题：用户能否在播放自己的视频流的同时向外界提供视频服务，如果能，理论上所能同时服务的最大用户数 n_{max} 是多少？

如果一个用户同时向外界提供了 n 个播放流，设每个流的播放速率为 $R_1, R_2, R_3, \cdots, R_n$，并用 $R^{avg}=\frac{\sum_{i=1}^{n} R_i}{n}$ 代表平均播放速率。

假设网络带宽和机器的处理能力均足够，则 n_{max} 受限于磁盘的最大数据提供能力 D_{max}，如下所示：

$D_{max} \geqslant n_{max} R^{avg}$，所以

$$n_{max}=\left\lfloor \frac{D_{max}}{R^{avg}} \right\rfloor \quad (1)$$

假设磁盘的数据提供能力和机器的处理能力均足够，则 n_{max} 受限于网络最大带宽 N_{max}，如下所示：

$N_{max} \geqslant N_{in}+N_{out}=R_{play}+n_{max}R^{avg}$，所以

$$n_{max}=\left\lfloor \frac{N_{max}-R_{play}}{R^{avg}} \right\rfloor \quad (2)$$

其中 N_{in} 是下行带宽，N_{out} 是上行带宽，R_{play} 是本机播放的速率。

假设磁盘的数据提供能力和网络带宽均足够，则 n_{max} 受限于机器的处理能力 P_{max}，如下所示：

$P_{max} \geqslant P_{play}+P_{serve}=R_{play}P_{ph}+n_{max}R^{avg}P_{ph}$，所以

$$n_{max}=\left\lfloor \frac{P_{max}-R_{play}P_{pk}}{R^{avg}P_{sk}} \right\rfloor \quad (3)$$

其中 P_{play}是本机播放视频流所需的处理能力，P_{serve}是向外界提供服务所需要的处理能力，P_{pk}是播放单位数据所需要的处理能力，P_{sk}是提供单位数据服务所需要的处理能力。

实际上，一台机器的磁盘数据提供能力、网络带宽和处理能力均有一定限制，所以 n_{max}同时受限于(1)、(2)和(3)式，故

$$n_{max}=\min\left\{\left\lfloor \frac{D_{max}}{R^{avg}} \right\rfloor,\left\lfloor \frac{N_{max}-R_{play}}{R^{avg}} \right\rfloor,\left\lfloor \frac{P_{max}-R_{play}P_{pk}}{R^{avg}P_{sk}} \right\rfloor\right\} \quad (4)$$

这就是一个用户在自己播放视频流的同时所能同时服务的最大用户数。在下面的实验分析中就会实际用到(式4)。

实验结果

下面以实验手段探测硬盘、网络和处理器三种瓶颈的存在。实验环境由两台性能差别很大的机器构成，机器 A1400为 AMD Thunderbird 1.4G CPU、512M DDR 内存，机器 P233为 Pentium 233 CPU、256M SDRAM 内存。两机以10M 以太网连接，为防止干扰，该网与其他网络隔离。机器对外界的视频服务能力以单位时间传输文件的总数据量来度量。由于视频服务的流式数据块的大小一般都是固定的，故用大小为100k 的文件来模拟数据块。

首先测定最大网络带宽 N_{max}。当所有数据都在内存中时，磁盘瓶颈得以消除，处理器瓶颈也可以忽略，这时测得两机间每分钟能传 68MB 数据（$N_{max}=68*8*1024/60=9284.27$kbps），接近于10M 以太网的极限值。

然后分别用 A1400和 P233作服务方(另一台作客户方)，测定服务方在自身不播放或播放不同速率视频流的情况下，能够向外提供的最大数据流量，并观察提供服务对自身播放质量的影响。结果如下表所示：

		不播放	播放 56kbps 视频流	播放 100kbps 视频流	播放 300kbps 视频流	播放 500kbps 视频流
A1400作 Server	1分钟传输的数据量(MB)	36.1	36.1	36.1	35.7	35.6
	对自身播放质量的影响		不影响	不影响	不影响	不影响
P233作 Server	1分钟传输的数据量(MB)	37.3	37.2	37.2	37.2	36.1
	对自身播放质量的影响		不影响	不影响	播放出现停顿	播放出现严重停顿

从表中可以看出，无论是 A1400还是 P233，1分钟对外传输数据均为36MB 上下，基本受限于磁盘的数据提供能力（$D_{max}=36*8*1024/60=4915.2$kbps），与 CPU 的处理能力和自身播放速率关系都不大。这说明，不同档次的计算机是比较容易保障对外服务的质量的。然而，当用 P233作服务方且自身播放速率为300Kbps 和500kbps 时，再对外服务就造成播放出现停顿和严重停顿(而如果不对外提供服务，播放300kbps 乃至500kbps 的视频流是没有停顿的)。这说明，档次低的机器的自身播放能力受到 CPU 的处理能力的影响。

下面根据(4式)和实测参数计算 A1400和 P233的 n_{max}值。为了简化，假设自身播放的视频流速率和对外服务的每条视频流速率均相等，例如 $R_{play}=R^{avg}=100$kbps。虽然处理器的处理能力对服务的影响甚微，却对自身的播放质量影响甚大，所以计算 n_{max}时要剔除影响自身播放质量的播放速率。

	自身播放及提供服务的每条视频流速率			
	56kbps	100kbps	300kbps	500kbps
受限磁盘：$\left\lfloor \frac{D_{max}}{R^{avg}} \right\rfloor$	87	49	16	9
受限网络：$\left\lfloor \frac{N_{max}-R_{play}}{R^{avg}} \right\rfloor$	164	91	29	17
A1400的 n_{max}	87	49	16	9
P233的 n_{max}	87	49	不可行	不可行

由此可见，如果网络净带宽达到10M，一台普通 PC 在本机播放视频流的情况下，还能同时为数十台其他 PC 提供视频流服务。当然，性能较差的 PC 如果要播放较高质量的视频流，再向外界提供服务将会影响自身播放质量。这个实验表明在网络带宽有保障的环境下，P2P 分布式 VOD 系统是可行的。当网络带宽有限时，例如只有512kbps 时，则 n_{max}只会受限于网络带宽，这时，播放300kbps 和500kbps 视频流的同时再提供相同质量的服务是不可能的，但如果播放56kbps 和100kbps 的视频流，还分别可以向外提供8路和4路同样质量的视频服务。需要指出的是，ADSL 是个特例，它的下行和上行速率差别很大；目前电信部门提供的 ADSL 服务下行一般达到512kbps，而上行只有50～60kbps。因此用户可以享受速率高达300kbps～500kbps 的视频流，却只能提供一路低速率的视频服务。

参考文献

1 Peer-to-Peer Working Group - Home. http://www.peer-to-peer-wg.org/index.html

2 Paulson L D. Microsoft, sun announce P2P technologies. Computer,2001,34(9):21

3 Flammia G. Peer-to-peer is not for everyone. IEEE Intelligent Systems:[see also IEEE Expert],2001,16(3): 78～79

4 Fox G. Peer-to-peer networks. Computing in Science & Engineering,2001,3(3):75～77

5 Parameswaran M,Susarla A, Whinston A B. P2P networking: an information sharing alternative. Computer , 2001,34(7): 31～38

6 Napster HomePage. http://www.napster.com

7 .NET My Services homepage (formerly code-named Hailstorm) http: // www. microsoft. com/net/netmyservices. asp , http: // www. microsoft. com/myservices/services/userexperiences. asp, http: //www. microsoft. com/net/hailstorm. asp

8 Jxta HomePage http://www.jxta.org

9 MacFarlane J. PCs enlisted to cure cancer. Nature Medicine,2001, 7:517

10 Schreiner K. Distributed projects tackle protein mystery. Computing in Science & Engineering ,2001,3(1):13～16

11 Tilley S. Spreading knowledge about Gnutella: a case study in understanding net-centric applications. Program Comprehension, 2001. IWPC. 2001. In: Proc. 9th Intl. Workshop on , 2001. 189～198

12 Chen Qiming,Hsu M. Inter-enterprise collaborative business process management. Data Engineering, 2001. In:Proceedings. 17th Intl. Conf. on ,2001. 253～260

13 Fox F,Gannon D. Computational grids. Computing in Science & Engineering,2001,3(4):74～77

14 Imesh HomePage, http://www.imesh.com

15 MojoNation HomePage, http://www.mojonation.net

16 Freenet HomePage, http://freenet.sourceforge.net

17 Gnutella HomePage, http://gnutella.wego.com

18 OpenP2P HomePage, www.openp2p.com

19 SETI@home Project, http://setiathome.ssl.berkeley.edu