对等网络的抖动特性研究综述

付志鹏[1),2)] 王怀民[1),2)] 史殿习[1),2)] 邹 鹏[1)]

[1)](国防科学技术大学计算机学院 长沙 410073)

[2)](国防科学技术大学并行与分布处理国家重点实验室 长沙 410073)

摘 要 对等网络应用是目前互联网上最主要的应用之一,但是它的性能受到抖动特性——节点频繁加入和退出网络的影响.文章在系统简述抖动的由来、定义及其对 P2P 系统性能影响的基础上,详细介绍抖动的统计特性研究,发现如节点的会话时长一般服从重尾分布等的一些动态规律;详述抖动的测量方法研究,针对被动监测,主动监测以及抽样测量等阐述各自的优缺点,并说明相应的改进方法来提高网络测量的精度;详述为减少抖动影响的应对策略研究,在邻居选择、失效恢复、副本维护、连接生命周期维护等方面说明各应对策略的功能和优缺点,并针对各个方面分别阐述自己的看法.最后对未来的研究趋势进行了总结和展望.

关键词 对等网络;抖动;统计特性;测量方法;应对策略

中图法分类号 TP393 **DOI 号**: 10.3724/SP.J.1016.2011.01563

A Review on the Churn Character of P2P Networks

FU Zhi-Peng[1),2)] WANG Huai-Min[1),2)] SHI Dian-Xi[1),2)] ZOU Peng[1)]

[1)](*School of Computer Science, National University of Defense Technology, Changsha* 410073)

[2)](*National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha* 410073)

Abstract P2P application is one of the most popular applications in the Internet, but its performance is badly influenced by the churn character—the nodes continuous arrival and departure. On the basis of introducing the origin of the churn, the definition of it and its influences to the P2P system performance, we minutely describe the research on the statistical properties, and discover some dynamic law including nodes' session time generally obey the heavy-tailed distribution. Then we detail the measurement methods. For passive monitoring, active monitoring, sample and so on, we present the advantages and disadvantages of each method, and explain the corresponding improving method to enhance the accuracy of the measurement. After that we elaborate the resilience strategies to the churn. Towards the neighbor selection, failure recovery, replica maintenance, link lifetime maintenance and so on, we present also the advantage and disadvantage of each strategy, and present our view to each of aspects. Finally, the summary and prospect of the future research are given.

Keywords P2P; churn; statistical properties; measurement methods; resilience

---

收稿日期:2011-02-19;最终修改稿收到日期:2011-05-18.本课题得到国家杰出青年科学基金(60625203)、国家"九七三"重点基础研究发展规划项目基金(2011CB302600)资助.**付志鹏**,男,1981 年生,博士研究生,主要研究方向为对等网络、分布计算. E-mail: zhipengfu518@gmail.com. **王怀民**,男,1962 年生,博士,教授,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为分布计算中间件、软件 Agent、网络与信息安全. **史殿习**,男,1966 年生,博士,教授,主要研究领域为分布计算、中间件、普适计算. **邹 鹏**,男,1957 年生,博士,教授,博士生导师,研究领域为分布计算、操作系统.

① Emule. http://www.emule.com
② Pptv. http://www.pptv.com
③ Overnet. http://zh.wikipedia.org/zh-cn/Overnet

1 引 言

P2P 应用是目前互联网上最主要的应用之一，遍及文件共享、实时通信、协同计算、流媒体传输等. P2P 软件如 BitTorrent[1]、eMule①、PPTV② 等都已经有几亿的用户群体，同时上线人数达到数百万规模，占据了目前互联网流量约 73.79% 的带宽[2]. 但是 P2P 的性能受到网络动态性（比如抖动，churn）严重制约.

从 P2P 网络诞生开始，churn 就一直伴随其左右. 2000 年到 2004 年间，研究人员通过各种手段对 Napster[3]、Gnutella[4]、Overnet③ 等 P2P 网络进行测量[5-9]，发现网络中节点的平均会话时长一般都比较短，理论分析和实验验证表明，节点的频繁加入和退出对网络的拓扑一致性、维护开销、性能以及可靠性等都会产生影响. Churn 的存在制约了 P2P 的发展及其推广应用. 有越来越多的研究者关注并参与研究. 在 SIGCOMM、INFOCOM、ICNP、IPTPS、IEEE P2P 等著名国际会议以及 IEEE/ACM Transactions on Networking 等期刊都收录了多篇关于 churn 的文章. 这些文章从 churn 的统计特性、测量方法以及应对策略等方面对 churn 进行全方位、多角度的研究，提出了很多宝贵和富有建设性的意见.

从实际应用的角度出发，针对目前唯一在现实中大量应用的结构化 P2P 网络——Kademlia[10] 进行 churn 特性研究是近几年来该方面的特点，研究人员通过网络测量、理论分析、实验验证等手段对其应用软件 eMule、BitTorrent 进行研究，取得了丰富的研究成果.

文献[11]对 churn 进行了部分阐述，本文致力于对其进行进一步的深化和补充.

本文第 2 节阐述 churn 的定义及其对 P2P 网络的影响；第 3 节详述 churn 的统计特性，随后阐述 churn 的测量方法；第 5 节阐述 churn 的应对策略；文章最后讨论 churn 的研究发展方向.

2 Churn 简介

2.1 Churn 的由来

Churn 一词的产生由来已久[12]. 随后该词被扩展到多个领域. 它表示前面的状态还未调整好，下一个活动又添加进来，系统不停地抖动，并且一直延续下去. P2P 网络中的 churn 是指，当一个节点加入或退出网络后，网络需要调整来适应这种变化（包括与其相连的节点信息的更新，与节点相关的资源信息的更新等），而不等网络调整好，下一个节点又加入进来（或退出），又需要调整，……，使得网络处于不停的抖动状态.

在 P2P 网络诞生的初期，人们非常渴望知道 P2P 网络的一些静态和动态特点，比如规模到底有多大？网络节点的活动是否有规律？等等. 马萨诸塞州立大学的 Chu 等人[6] 于 2000 年 12 月 21 日～2001 年 2 月 3 日对 Napster 及 2002 年 2 月 24 日～3 月 25 日对 Gnutella 的监测表明，有 31% 的节点的会话时长仅为 10 min 左右，仅有不到 20% 的节点的会话时长超过 2 小时. 华盛顿州立大学的 Saroiu 等人[7] 同样对 Napster 和 Gnutella 进行监测，结果表明，网络中有 50% 的节点的会话时长少于 60min. 加州大学圣地亚哥分校的 Bhagwan 等人[5] 于 2003 年 1 月 14 日～1 月 28 日对 Overnet 网络的监测表明，50% 的节点只有 0.3 的可用性①.

在 2003 年 11 月加州大学伯克利分校的技术报告中，Rhea 等人[8-9] 总结了 2000 年～2003 年间研究人员对 P2P 网络的监测结果，表明 P2P 网络中节点加入和退出非常频繁，平均在线时长由几分钟到一小时不等，并且该现象对网络的性能和可靠性产生严重影响，必须引起 P2P 网络研究人员的足够重视. 他借鉴工业领域以及经济领域中 churn 的定义，首次将网络中节点频繁地加入和退出网络也定义为 churn，得到学术圈同行们的广泛认同，从此，学术界统一将该现象称为 churn.

2.2 Churn 的定义

虽然 Rhea 在文献[8]中给出了第一个定义：节点加入和退出的持续过程（the continuous process of node arrival and departure）. 但是随后，众多研究者对 churn 给出了自己的理解和定义，比较著名的有：

麻省理工大学的 Li 等人[13] 将 churn 定义为：成员关系的持续改变（continuous changes in memberships）.

加州大学伯克利分校的 Godfrey 等人[14] 将 churn 定义：由于加入、正常离开、失效导致的一系列参与节点的改变（change in the set of participating nodes due to joins, graceful leaves, and failures）.

① 可用性用响应次数和探测次数的比值来计算，比如探针探测 10 次，有 3 次响应，则可用性为 0.3.

俄勒冈州立大学的 Stutzbach 等人[15]将 churn 定义为：成千上万的节点由于独立地加入和退出所产生的叠加效果(the collective effect created by the independent arrival and departure of thousands (or millions) of peers).

从这些定义可以看出，虽然它们的说法各不相同，但其内涵基本一致，churn 的概念有两个关键要素：

(1) churn 是由于节点加入或退出引起的，没有节点的加入或退出，就不会产生 churn 现象.

(2) churn 不是描述单个节点的行为，而是描述整个网络的动态特点，它是由于单个节点的行为而产生的整体结果. 单个节点的一次加入或退出并不能代表什么，并且可以通过更新路由表、邻居表等，很快调整过来. 而对整个网络而言，不断地有节点加入，同时不断地有节点退出网络，并一直延续下去，直到永远.

一直以来，都很难找到一个相应的中文词语来准确表达 churn 的意思，翻译成扰动、抖动、波动、颤动等的都有. 目前来说，大多数人趋向于翻译成抖动，本文也采用大众的看法，用抖动来表达.

2.3　Churn 对 P2P 网络的影响

当 churn 产生时，就会对网络的正常运行产生影响，使其变得非常低效. 总结来说，churn 对 P2P 网络的影响主要包括如下几个方面：

(1) churn 对 P2P 网络的拓扑一致性产生影响.

P2P 网络的拓扑结构用数学方法描述为用点(表示网络中的一台台计算机)和边(代表计算机之间的连接)连接而成的拓扑图. 当节点加入和退出网络比较频繁时，节点的邻居往往经常变化，这使得表面上存在于邻居表中的节点实际上由于失效或下线而无法连接，从而产生网络的拓扑结构和实际不一致的情况. 这将导致节点的连接失效，数据不可达等问题，严重影响网络的性能.

(2) churn 使得 P2P 网络的维护代价提高.

在 P2P 网络中，每个节点都保存有一张路由表，用于存储和自己相连的邻居信息. 当节点频繁加入和退出网络时，需要定期发送消息来检查确认邻居节点是否在线，当网络规模比较庞大时，这种消息的数量是很惊人的. 假设网络节点数目为 N，每个节点的路由表大小为 $O(\log N)$，每隔 30 s 发出一条大小为 $2k$ 的心跳信息，则整个网络每分钟产生的负载为 $4k\times N\times O(\log N)$ 的消息量，对于一个具有 100 万节点的网络，其数量大约为 80 G(事实上在 Gnutella 网络初期，其网络维护消息的流量占其总流量的 80%左右[16]). 再加上在发现邻居节点失效时，需要及时查找新的邻居来保证路由表的完整性. 这些由于 churn 带来的维护消息加重了网络负担，提高了网络维护代价.

(3) churn 对 P2P 网络的性能产生影响.

衡量 P2P 网络的性能指标有多个，其中包括路由效率、查找延时、数据传输率等. 而 churn 对这些指标都有影响. 在 P2P 网络协议的设计中，网络路由、资源查找、数据传输都以节点邻居一直保持不变这一假设为前提. 但在实际网络的路由传递过程中，下一节点可能失效，导致路由不可达，而源节点可能并不知道而盲目等待，既浪费时间，又降低效率. 资源查找时，由于节点的加入和退出，导致有些资源可能根本不在网络中或者需要很长时间才能找到，从而增大查找延时. 在数据传输的整个过程中，并不能保证传输路径上的每个节点一直在线，当节点频繁加入和退出网络时，若保证数据传输不中断，网络需选择其它路径来进行传输，降低了数据传输的效率.

(4) churn 对 P2P 网络的可靠性产生影响.

P2P 网络的可靠性是指 P2P 网络完成任务的正确性. 网络路由不可达，数据不能传输到指定的节点，传输的数据产生错误、不完整，在网络中明明标识存在的资源却找不到等等，都是不可靠的表现. 在 churn 下，P2P 网络可能产生网络分割，某些节点被孤立，使消息传递不到给该节点从而使得网络不可靠. 在进行资源的查找时，有些已经发布在网络上的资源的源节点可能已经失效，从而导致资源不可靠. 在数据传输过程中，由于 churn，导致传输过程中的节点失效，产生数据不可达、丢包等现象，导致传输不可靠.

总之，churn 对 P2P 网络的正常运行会产生极大影响，研究 churn，获得网络节点的活动规律，分析这些规律对网络的影响，并据此提出一些切实有效的应对策略来减小这些影响，对提高 P2P 网络的运行效率，促进 P2P 网络的推广使用都有重要作用.

3　Churn 的统计特性研究

研究 churn 统计特性的目的是为了获得 P2P 网络的动态规律.

表现 churn 统计规律的一些主要参数包括节点会话时长(session time)、在线时长(uptime)、剩余

时长(remaining uptime)、生命时长(lifetime)等.下面分别对其进行简述.

3.1 会话时长研究

会话时长是指单个节点的一次加入—参与—退出网络的活动周期所经过的时间,它用节点离开网络的时刻减去节点加入网络的时刻来计算.从 2002 年开始,一系列的研究人员对不同的 P2P 网络进行了统计监测,其结果如表 1 所示.

表 1 会话时长监测结果总结

发表年份	研究者	网络名称	会话时长分布
2002	Chu 等人	Napster、Gnutella	二次对数分布[6]
2003	Gummadi 等人	KaZaA	重尾分布[17]
2003	Bustamante 等人	Gnutella	Pareto 分布[18]
2005	Stutzbach 等人	Gnutella、BitTorrent、Kad	幂率分布[19]
2006	Stutzbach 等人	Gnutella、BitTorrent、Kad	Weibull 或 log-normal 分布[15]
2009	Steiner 等人	Kad	Weibull 分布[20]

会话时长到底服从什么分布,目前尚无定论,每种说法都有自己的道理.本文认为,由于网络的动态性,会话时长也是一个动态变化的过程,在某个时间段,它可能用 a 分布能够很好地拟合,但在另一个时间段(比如热点爆发),a 分布不能拟合,而它却能用 b 分布很好拟合,热点过后可能用 c 分布又拟合得很好.精确描述会话时长的分布比较困难,但总结网络活动规律对于指导网络和节点的行为,却具有很重要的参考价值.目前一个普遍的共识是,节点的会话时长一般服从重尾(heavy-tail)分布,比如 Pareto 分布和 Weibull 分布.

3.2 在线时长研究

P2P 中 churn 的统计特性进行研究的另一个重要参数就是在线时长,它是用目前在线节点的当前时刻减去节点加入网络的时刻来计算.Sariou 等人[21]对 Napster、Gnutella 网络进行研究后发现,节点在线时长服从 Poisson 分布.而 Stutzbach 等人[15]对 Gnutella、BitTorrent、Kademlia 网络进行研究后表明,在线时长服从幂律分布.

3.3 剩余时长研究

研究会话时长和在线时长的意义在于预测剩余时长,它是指从节点当前时刻到节点将来下线所剩余的时间.即剩余时长=会话时长−在线时长.通过预测剩余时长可以为网络选择节点提供依据.

Yao 等人[22]通过建立一个隔离模型来研究剩余时长:对每一个加入系统的用户 v 都分配一个随机的会话时长 L,L 的分布函数已知.当加入系统时,v 找到 k 个初始邻居,然后持续监测它们的状态.当有邻居失效时,都将随机选择系统中在线的节点替代(任何时刻,邻居只有两个状态:活着(ON)或者死亡(OFF).ON 的持续时间 R 为邻居的剩余时长,OFF 的持续时间 S 为查找替代节点的延迟),然后通过构造一个持续时间的 Markov 链来跟踪 v 的出度.文章通过一系列的理论推导、实验,得出结论:产生剩余时长长的策略可以确保系统具有较低的隔离频率和较高的抗 churn 的适应性(resilience).为此,在节点选择时,选择剩余时长长的节点可以使系统更稳定.但是节点的剩余时长是一个未来的值,实际中不可能知道.而会话时长具有重尾分布的特点告诉我们,在线时长比较长的节点,其剩余时长也往往比较长.因此,很多研究者据此对剩余时长进行预测.

Stutzbach 等人[15]通过测量实验指出可以对 Gnutella 和 Kademlia 的剩余时长进行较好的预测,而 BitTorrent 不能.Steiner 等人[20]同样通过对监测数据的分析指出,一个上线达到 1000 min 的节点,将有 1500 min 的剩余时长,只有 20% 的在线时长为 2 h 的节点将继续在线 24 h.

3.4 生命时长研究

节点的生命时长是指用户第一次使用 P2P 软件上线到最后一次卸载终端软件离开网络所经过的时间.在一些文献中,节点生命时长是指节点的会话时长.由于节点此次下线后,不能确定下次是否会上线,在进行生命时长测量时,往往具有不确定性.同时,由于生命时长的跨度往往比较长,而一般的网络实验却不会持续太长时间.这些因素都导致节点生命时长的测量存在一定的难度.Steiner 等人[20]针对 KAD 网络进行了 6 个月的监测发现,中国节点和欧洲节点的生命时长有大的不同,超过 1/3 的中国节点在一天之后消失,只有 10% 的中国节点的生命时长超过 150 天.而接近 40% 的欧洲节点的生命时长超过 150 天.同时,文章还研究了生命时长和会话次数之间的关系,分析表明节点的生命时长强烈地依赖节点重连系统的次数.大概 30% 的中国节点其生命时长只使用一次会话,而欧洲节点只有 5%.

3.5 针对 KAD 网络的统计特性研究

早期针对 P2P 网络的测量大多是基于非结构化 P2P 网络来进行,基于结构化 P2P 网络的非常少,而且结构化 P2P 网络大多只存在于实验室,对这些网络的测量都以模拟为主.实际使用得非常少,

而 Kademlia 网络是实际应用非常广泛的网络，在目前的 BitTorrent 系列软件，eMule 系列软件中都使用它，拥有大量的用户群体. 为此，针对基于 Kademlia 网络的测量具有较强的应用价值，是近年来国内外 P2P 网络测量研究的热点. 为此本文单独拿出一小节来进行阐述，逻辑结构上可能和前面的不一致.

Steiner 等人[20]针对 KAD 网络的监测表明：节点的会话时长服从重尾分布，至少有 0.1%的会话超过 1 周，并且被监测到的最长的会话为 78 天，最适合的分布函数为韦伯(WeiBull)分布. 文章还对会话之间的相关性进行了分析，结果发现以前的会话和下一次会话之间的相关度只有 0.15，即通过以前的会话来预测将来的会话并不准确. 但是当只考虑超过一天的长时长会话时，它们的正相关达到 0.85. 通过监测还发现，中国的节点比欧洲节点每日花非常少的时间连接到网络上，欧洲有 40%的节点的日平均上线时间为 5 h，20%的节点超过 10 h，而且大部分节点每天的可用性变化非常大，通过已有信息预测每天的可用性比较困难. 在欧洲，每个节点平均拥有 18 个 IP 地址，而在中国每个节点拥有 4 个 IP 地址. 80%的中国节点只有一个 IP 地址，这缘于它们的生命时长比其他国家的短很多. 用 KAD ID 来唯一标识节点并不准确，有些节点的 ID 是变化的，通过监测法国 ADSL 客户端发现，大约 20%的节点为每个新会话改变他们的 KAD ID，有些甚至在一个会话中改变它. 地理分布上，节点比例最高的大陆为欧洲，节点数量最大的国家为中国，只有少于 15%的节点在美洲(美国、加拿大、南美)，任何时候网络中都有接近 25%的节点来自中国.

Memon 等人[23]针对 KAD 网络中的消息通信进行监测，在 2008 年 5 月到 2008 年 8 月间收集了 44 组高 6 位相同的 KAD 域(zone)之间的通信消息. 分析数据后的结果显示，10%已发布内容 ID 每分钟的请求速率为 0.1 条以上，而 0.1%已发布内容 ID 的请求速率每分钟超过 30 条. 而内容发布速率与内容搜索速率的差别非常大，一些关键词的发布速率超过 100 条/min，而相应的，被监测到的搜索请求最高时才低于 2 条/min. 这一部分原因在于内容发布是系统自动产生而搜索却是用户行为. 15%从没发布过的文件得到搜索，而 60%已发布的文件却从未被搜索. 有高达 95%的已发布的关键词从未被搜索，由此可以看出和文件相关的多个关键词只有很少一部分被真正使用.

4　Churn 的抓取和测量方法的研究

随着研究人员对测量的深入以及对多种数据抓取和测量方法所得到的数据统计结果的分析比较，人们发现，早期的很多数据抓取和测量方法存在偏差，由此得到的很多数据都不可靠甚至是错误的. 因此，大概从 2006 年以后，研究人员越来越关注对数据抓取和网络测量方法进行改进的研究.

对 P2P 网络测量方法并没有严格的分类，Stutzbach 等人[15]将其分为被动监测(passive monitoring)和主动监测(active monitoring)两种. 而后来他又将其进一步分为被动监测、参与(participate)、爬测(crawl)、抽样(sample)和中心化(centralize)等[24]. 本文认为主动监测和被动监测是从数据获取方式的角度去分类. 参与是指每个参与其中的节点都记录本地自己的统计信息，并据此来进行分析的方法，可以认为是被动监测的一种. 抽样是从数据筛选的角度来看的，在主动监测和被动监测中都可以结合使用. 中心化主要是通过获取中心服务器的数据来进行分析的测量方法，在主动监测和被动监测中也都可以使用. 为此，本文还是将测量方法分为主动监测和被动监测两类，并分别进行详述. 抽样测量是在网络规模非常庞大时对网络进行测量分析的主流方法，经常用于对非结构化 P2P 网络的测量分析，为此本文也单独拿出一小节来对其进行详述. 针对目前大规模使用的 BitTorrent、eMule 进行动态模拟和测量具有很强的应用价值，本节最后也对其进行简述.

4.1　被动监测

被动监测是指在 P2P 网络中选取足够数量的节点(最好是位于骨干网络中)作为监测节点，记录所有经过该节点的消息流的相关信息.

4.1.1　方法评价

该方法占用 CPU 比较少，监测过程中基本上不占用通信带宽，因此不会对 P2P 网络的正常运行产生较大的影响. 被动监测本身的缺陷使得该方法存在较大的不足：

(1) 监测到的节点数目有限. 它只能监测那些发送消息且消息恰好经过被监测节点的节点，对于那些存在网络中但很少甚至不发送消息的大量边缘节点是无法监测到的. 同时，一些边缘节点(地理相近的园区节点、社区节点)在小范围内聚集成簇，在簇内如果没有监测节点，它们之间的通信也是无法监测到的.

(2)监测到的信息具有较大的误差.假设有一个源节点 A 在 T_0 时刻上线,但是由于它不发送消息(比如 Bittorrent 中不上传下载任何数据的节点)或者在监测系统启动前已经初始化完毕,监测系统发现不了它上线,经过时间 t_1 后,A 在 T_1 时刻向网络中发送消息,此时监测系统才发现 A 上线,经过 t_2 时间后,A 在 T_2 时刻接收到最后一条消息,此后不再发送和接收任何消息.然后在 T_3 时刻节点 A 下线.节点 A 实际存在于网络的时间为 $T_0 \sim T_3$,而监测系统发现 A 的上线时间却为 $T_1 \sim T_2$.对于 $T_0 \sim T_1$,$T_2 \sim T_3$ 这两个时间段 A 是否存在于网络中,存在多长时间,监测系统是无法获知的.这肯定有较大的误差.

(3)通过消息流来判定节点可能存在的误差.消息流主要通过 IP 地址和端口号来标识一个节点,但是研究表明很多节点的 IP 地址和端口号是动态变化的,对于那些使用动态 IP 地址的用户来说,它们可能被标识为多个节点.而有些小局域网可能采用一个 IP 地址和端口号上网,因此可能有多个用户被标识为同一个节点.

4.1.2 适用场景

虽然被动监测对获取网络的节点数目以及节点上线下线时间存在误差,但是它对网络通信特点的监测却非常管用,特别是集中式和具有超级节点的分布式系统中的网络通信的监测.

Sen 等人[25]采用被动监测方法来对网络通信进行分析.他通过在第一级网络服务提供商(ISP)的骨干网的多个路由器上部署监测点,收集通过这些监测点的所有流信息(包括数据流和消息流),并对这些流信息在 3 个不同的粒度层级上进行分析:IP 地址级、网络前缀(Network Prefix)级、自治系统 AS(Autonomous System)级,从而获取网络中消息流的特点.文章通过对收集的 2001 年 9 月到 11 月间在 Gnutella、FastTrack、DirectConnect 网络的通信流分析发现,网络中通信最活跃的时间发生在傍晚和午夜.凌晨 5 点过后,网络通信量随着时间慢慢变少,到下午一二点时达到最低值.60%的 IP 地址,40%的网络前缀,30%的 AS 每天在网络中所驻的时间少于 10 min,超过 20%的网络连接持续时间只有 1 min 甚至更少.65%的 IP 地址在 FastTrack 网络中只驻留一次.

4.1.3 方法改进

Memon 等人[23]针对被动监测的不足进行改进,并用改进后方法对 KAD 网络中的消息通信进行监测.

针对被动监测中插入少量监测节点不能获取足够详细和全面的通信消息,而插入过多的监测节点又会干扰 DHT 网络的正常运行这一问题,Memon 等人提出并实现了一种高度并行的、可扩展的被动监测技术——Montra.其关键思想是让监测节点的可见度尽量小,从而减小监测节点对系统的干扰,并降低对监测节点的资源要求.它通过如下策略来进行:(1)将监测节点 P_m 的 ID 设置为目标节点 P_t 的 ID 与 1 的异或值,即 $ID(P_m)=ID(P_t)\oplus 1$,从而保证 P_m 为 P_t 距离最近的节点,因此根据消息应发送给距目标 ID 最近的多个节点的原则,发送给 P_t 的消息,P_t 一定也会发给 P_m,从而可以监测所有经过 P_t 的消息.(2)Montra 通过最小化监测节点的可见性来尽量减小监测给网络带来的干扰,它使监测节点 P_m 只对目标节点 P_t 可见,而对其它节点均不可见,并且 P_m 只对 P_t 的消息响应,而对其它消息均忽略,同时 P_m 不保存任何网络内容.通过这两个步骤,使得 P_m 可以在网络中大规模部署而不增加网络的负担.在 KAD 网络上的运行显示,Montra 可以同时监测 32 000 个 KAD 节点上的消息通信,而丢包的概率为 0.009%,它可以抓取 90%的查询消息并定位 90%的目标节点.

4.2 主动监测

和被动监测不一样的是,主动监测是节点通过探测主动获取邻居节点的信息,然后通过邻居节点获得邻居的邻居节点的信息,如此循环.这一过程主要通过在网络中运行一定数目的爬虫(crawler)来主动抓取网络的快照信息获得.

4.2.1 方法评价

该方法需要占用爬虫所在机器的大量 CPU 时间和存储空间以及非常高的网络带宽.同时,主动监测对 P2P 网络的正常运行可能存在干扰,一些正在进行业务处理或数据传输服务的节点由于 CPU 或网络带宽被爬虫程序占用而被迫延迟或中断.虽然如此,和被动监测相比,主动监测的优点是非常明显的.

(1)监测到的信息具有较高的准确性.爬虫程序每隔一个单位时间 Δ 就抓取网络的快照一次,假设节点在 T_k 时刻的快照中没抓取到,在下一个快照 T_{k+1} 时刻抓取到了,则节点的上线时间应介于 $T_k \sim T_{k+1}$ 之间,误差不会超过 Δ,当 Δ 较小时,它的准确度是比较高的,文献[26]说明,抓取整个网络需要 3 min,而文献[20]说明抓取网络中高 8 位相同的

zone crawl 只需 2.5 s，当间隔时间 Δ 取这些值时，可以保证主动监测比被动监测具有更高的准确性.

(2)能够比被动监测抓取到更多的节点，使网络快照更接近于真实的网络. 当爬虫程序经过多次迭代后，如果发现此次迭代抓取到的节点绝大多数是已经获得的节点时，根据小世界模型理论，可以认为，我们已经抓取到网络中绝大多数的节点，此时抓取到的网络快照就比较接近真实的网络.

4.2.2　方法改进

当然，主动监测方法也有很多地方需要改进. Steiner 等人[20,27]针对以往主动监测中需要多台主机并行执行，消耗过多时间用于多台主机间的同步问题，提出了一个快速定制爬虫——Blizzard. 它将爬虫程序只放在一台电脑上，该爬虫在初始运行时，就已经和数以百计的节点相连，并使用一个简单的宽度优先搜索来迭代询问邻居节点，通过这些已知节点来发现新的节点，对于每个节点应答，都确认并排除已发现的节点. 相比传统的主动监测方法，由于减少了同步通信问题，该爬虫能够快速高效地抓取网络信息. 但是该方法对软件、硬件以及环境都提出了更多的要求，初始时就必须和数以百计的节点相连，使节点必须处于网络的骨干位置，由于要在非常短的时间内和数以百万规模的网络节点通信，对机器的处理速度、存储空间以及网络带宽都提出了非常高的要求. 通过在法国和德国分别部署该爬虫来对 eDonkey 中的 KAD 网络进行监测，从 2006 年 9 月 23 日到 2007 年 3 月 20 日进行高 8 位相同的 zone crawl，表明每 2.5 s 就可以获得一次快照，具有较高的速度. 从 2007 年 3 月 20 日到 2008 年 5 月 25 日进行的 full crawl 表明，获取整个网络的快照需要 8 min，每次 crawl 可获得 3 百万到 4.3 百万个节点，其中前两百万个节点只需 1 min，同样具有较快的速度.

周模等人[28]根据 KAD 网络的特点，设计并实现了一个可扩展爬虫，该爬虫使用宽度优先搜索和查询迭代交互进行. 在爬虫工作时，有一个已知节点集 N_{know} 和结果节点集 N_{result}(初始时 N_{result} 为空)，对 N_{know} 中的每一个节点 ID_{know}，爬虫都通过公式 $ID_i = ID_{know} \oplus 2^{127-i}$ $(0 \le i \le 127)$ 来计算出需要在 ID_{know} 中查询的节点 ID，并向节点 ID_{know} 发出查询该 ID 的请求，节点 ID_{know} 收到请求后就会查找它的路由表第 i 个链表中所有离 ID_i 最近的节点列表，并将其返回给爬虫节点. 爬虫收到这些节点列表后，对每个节点首先查看它是否已在结果节点集 N_{know} 中，如果存在就放弃该节点，否则就将其加入到 N_{result} 中. 在爬虫向 ID_{know} 发送查询请求后，可以接着向下一个节点通过上面的步骤计算出新的节点 ID 然后向该节点发出查询请求，当对 N_{know} 中的节点都发送了查询路由表中第 i 个链表的请求后，第 i 次查询结束. 然后，爬虫重复上述步骤对 N_{know} 中的每个节点发出查询路由表中第 $i+1$ 个链表的请求. 直到 i 为 127 结束，则该轮爬虫迭代结束，然后将结果节点集 N_{result} 作为新的已知节点集 N_{know} 进行新一轮迭代. 为获得一个高效的可扩展爬虫，文章提出 3 个策略来提高效率：(1)减少每次迭代的轮数；(2)增加每一轮所获得的节点数目；(3)提高请求的发送频率. 文章通过对实验数据的分析，得到在一次迭代中哪几轮的查询能够获得非常多的不同节点，而哪几轮所获得的节点数目不多且与前面的重复，从而决定哪些轮非常重要，哪些可以剔除，进而减少每次迭代的轮数. 而留下来的那些轮，由于具有非常多的不同节点，等价于增加了每一轮所获得的节点数目. 从而据此来提高爬虫的效率.

Wang 等人[26]针对以往采用主动监测测量节点会话时长不准确的问题，提出自己的测量方法. 以往的测量方法中，当抓取快照的时间间隔 Δ 比较小时，会加重网络的负担，影响 P2P 网络的正常运行；当 Δ 较大时，就会有获取节点会话时长不准确问题. 对此，Wang 等人提出基于剩余时长的测量方法(Residual-Based Measurement)，它首先在 T_0 时刻首次获取网络快照的时候抓取到数目足够多的初始节点集 S_0，然后从 S_0 中随机选取 ε% 的节点作为跟踪节点集 S_1，随时跟踪 S_1 中每个节点的状态，直到节点下线或者达到实验时间 T 结束. 文章通过一系列的理论计算来证明其方法足够好，但是如何精确获取节点的下线时间却并未交待. 同时，文章给出的跟踪节点集具有较强的时间依赖性，只能是在首次抓取时恰好上线并被抓取到，才可能成为跟踪节点集的一员，其它时刻上线的节点无法被跟踪，因此对监测短会话时长节点是不利的. 文章最后采用该测量方法实现了一个针对 Gnutella 网络的爬虫——GnuSpider，该爬虫只需 3 min 就能覆盖整个网络，发现近 6.4 百万个用户，90% 的超级节点和叶节点都能在 100 s 内发现. 文章使用第一次抓取到的其中 46.8 万个应答超级节点为 S_0，取 $\varepsilon = 21.3$，即 $S_1 = S_0 \varepsilon\% = 10$ 万进行持续的 72 h 跟踪以获得其会话时长，其会话时长的误差控制在 3 min，即一次 crawl

过程,具有较高的精度.

4.3 抽样测量

和被动监测、主动监测不一样,抽样测量是从另一个的角度来看的一种网络测量方法,该方法可以和被动监测、主动监测结合起来使用.它是在实际网络中公平随机均匀地选择一些节点,将这些节点组成的抽样网络取代真实的网络来进行研究.由于真实的网络具有规模巨大和高度动态的特点,直接测量往往比较困难,并且代价比较高昂,抽样测量就成为一种可选的方案.

4.3.1 方法评价

和真实的网络测量相比,抽样测量大幅度减少了网络规模,降低了网络的复杂度,减小了测量的代价.但是,它毕竟不是真实的 P2P 网络,因此会使得网络的某些特性不明显甚至失去网络的某些特性.因此,如何保持真实网络的特性,同时在抽样时如何选取代表节点,是抽样测量需要特别注意的两个问题.该方法在以下方面存在不足:

(1)目前的抽样测量一般是基于静态图来进行的,而 P2P 网络是高度动态的.某时刻的抽样在下一时刻可能就会产生变化.虽然可以通过动态跟踪代表节点来解决动态性问题,但是由于节点离开,失效等行为时有发生,a 时刻具有代表性的节点在 b 时刻可能就不具有代表性.随着时间的推移,代表节点越来越少,抽样网络越来越不能代表真实的网络.

(2)代表节点的选择往往会产生偏差.在 P2P 网络特别是非结构化 P2P 网络中,处于中心位置的节点往往是那些节点度大、在线时间长、服务能力好的服务器节点,这些节点只占 P2P 网络中的很少一部分.绝大部分终端节点处于网络边缘,这些节点在线时间短、节点度很小.在抽样时,由于中心节点 24 h在线,什么时候抽样都可能被抽到,而终端节点往往上线时间短,错过了抽样时间就不可能被选中.同时,节点度大的中心节点被选中的概率也比较多,而终端节点由于节点度很小,抽样时走到该节点可能性也比较低,因此往往不容易被选中.

4.3.2 方法改进

Stutzbach 等人[29]针对非结构化 P2P 网络动态和异构对抽样测量带来的偏差,提出可回退受控随机走(Metropolized Random Walk with Backtracking,MRWB)的策略来进行抽样节点的选择.文章主要解决两个问题:如何处理网络动态性;在选择下一个节点作为抽样节点时,如何体现公平随机无偏性.对于处理动态性的问题,文中采用如下方法:

(1)对网络图加入时间目录:$G_t=\langle V_t,E_t\rangle$($G_t$ 为 t 时刻的网络图,V_t 为 t 时刻的节点集,E_t 为 t 时刻的边集);

(2)定义时间窗口:$[t_0,t_0+\Delta]$;

(3)该时间段出现的节点数为 $V_{t_0,t_0+\Delta}=\bigcup_{t=t_0}^{t_0+\Delta}V_t$;

(4)节点在不同时刻具有不同的性质,因此 $v_{i,t}$ 和 $v_{i,t'}$ 可以被重复选中;

(5)当测量窗口 Δ 足够小时,可以认为网络以及节点的性质改变较小.由此可以把这时间段中的网络图看成静态图来处理.

通过上面的 5 步,就可以将动态的网络图以时间为轴切成一个个静态图来分别进行抽样处理,从而简化了问题的复杂性.

在静态图中,抽样节点 x 在选择它的某一邻居作为下一个抽样节点时,按照下面的策略进行,首先平均随机地选择 x 的邻居 y,询问 y 的节点度从而计算出 $p_0=degree(x)/degree(y)$,然后生成一个概率随机数 $p(0\leq p\leq 1)$,如果 $p_0>p$,则 y 被选中作为 x 的下一跳,否则停留在 x 作为下一步.当 y 的节点度比较大时,p_0 就比较小,y 被选中的概率也比较小;但是,由于 y 的节点度比较大,当抽样走到 y 的任何一个邻居时,y 都有可能被作为下一个抽样节点而被选中,因此,节点度大的节点可能被抽到的次数就增多,从而弥补每次抽样时被选中的概率低的问题.该策略从一定程度上抑制了以往抽样偏向节点度大的节点而导致的偏差,在一定程度上实现了公平无偏性.但是,对于时间因素带来的偏差,文章并没有考虑.

文章通过从节点度抽样、会话时长抽样、查询延迟抽样等方面考虑针对不同会话时长分布(指数分布、Pareto 分布、Weibull 分布)的系统进行模拟实验,看在什么程度的抖动情况下该方法比较准确,在什么程度下该方法不准确.结果表明,当平均会话时长小于 2 min 时,该方法和希望结果存在明显差异,而对会话时长服从什么分布区别不大.通过跳数为 10000 步的模拟表明,该策略的精确度不受多步数的影响.模拟还显示,当网络最小节点度超过 3 时,该抽样方法不会产生明显偏差.文章最后采用该方法实现了一个网络抽样工具——ion-sampler.并从节点度和模拟跳数方面进行观察分析,结果显示相对其他抽样方法,该方法具有较高的精确度;相对于 full crawl,该抽样方法具有更短的抽样延时,而 full crawl 随着规模的增大其 crawl 延时呈线性增长,不

具备良好的可扩展性.

Rasti 等人[30]根据上面的结果更进一步提出了一种应答驱动的抽样 RDS(Respondent-Driven Sampling).它主要应用于为了获取网络中具有某种性质的节点所占的百分比所做的抽样.该抽样过程按照如下的方法进行：

(1)根据节点的某个属性 X(假设可取 m 个值)将网络划分成$\{R_1,\cdots,R_m\}$；

(2)将节点集也进行相应的划分$\{V_1,\cdots,V_m\}$,其中 $V_i=\{v\in V\mid X(v)\in R_i\}$；

(3)用于评估组 i 中节点占整个节点数的比例为 p_i；

(4)所有被访问的节点集合为 $T=\{t_1,t_2,\cdots,t_n\}$；

(5) $T_i=T\bigcap V_i$表示在组 i 中被访问的节点集；

(6) p_i表示评估组 i 中的节点的比例:则抽样值 $\hat{p}_i=\dfrac{S(\hat{I}_{V_i})}{S(\hat{I})}=\dfrac{\sum\limits_{v\in T_i}1/degree(v)}{\sum\limits_{u\in T}1/degree(u)}$ 和它的真实值 p_i 是一致的.

文章随后针对此方法进行了模拟实验,结果表明在静态图情况下,RDS 方法无论在网络抽样的精确度方面,还是在精确度随网络规模的线性增加方面,都比 MRWB 方法要好.对于动态图,当平均会话时长小于 5 min 时,节点频繁加入和退出导致该方法产生明显偏差,但是实际网络中的平均会话时长往往大于 5 min;同时,该方法也受网络最小节点度的影响,当最小节点度不小于 5 时,该方法的抽样精确度比较高,而当最小节点度小于 5 时,由于 churn 可能产生部分网络图不可访问而产生较大偏差.文章最后也将此方法融合在 ion-sampler 中,并采用 ion-sampler 的两种不同方法对 Gnutella 网络进行抽样测量,结果显示这两种方法都具有很高的精确度.

该抽样将网络按某一特性进行划分的方法为我们监测网络某种特性提供了新的思路.要想获得网络的完整全面的特征,该方法并不适合.同时,该文存在如下不足:(1)当我们观察的是网络的某一动态性质时,它会随着时间而改变,因此不好定在哪个划分中,比如要监测网络中 CPU 使用率为 60%以上的节点数目有多少?此时每台主机的 CPU 使用率是动态变化的,在某个时刻可能高于 60%,而在另外的时刻又可能低于 60%,从而不好确定节点属于哪一类.(2)该方法中回避了抽样节点如何选择的问题,只是简单说明是用随机走的方法选取节点,而随机走方法中节点的选取也有很多且精确度各不相同.

4.4　对大规模网络的模拟测量研究

由于真实 P2P 系统如 BitTorrent、eMule 等规模非常庞大、动态性强等特点,很难对其进行评估和预测,这给 ISP 及政府带来了许多麻烦.对这些系统模拟出尽量真实的环境,然后对其动态行为进行预测.对引导 BitTorrent、eMule 的行为发展具有重要的参考价值.

郑纬民等人[2]为此设计并实现了一个运行于服务器集群环境中的预测 P2P 行为的并行模拟器 AegeanSim.该模拟器的主要思想包括:模拟器的基本单元称为逻辑过程(Logical Process),它是可执行的数据结构而非线程或进程,用来模拟真实环境中的一个实体节点,每台模拟机器上有成千上万个逻辑过程,而模拟机器之间通过消息传送接口 MPI(Message Passing Interface)来实现模拟的并行和协同.为解决同步问题,AegeanSim 定义一个带时间戳的安全窗口,并严格控制其下限和上限.同时通过分组策略和简化接口来尽量模拟真实的 P2P 系统及提高模拟效率.文章通过模拟真实的 BitTorrent 系统并和 BT 研究进行比较,来验证模拟器的正确性.文章最后采用该模拟器对 tracker 断开时间对文件共享率的影响以及限制带宽对 BT 的影响来进行模拟,从而为 ISP 控制 BT 提供建议.

5　Churn 的应对策略研究

从发现 churn 会对 P2P 网络的正常运行产生各方面的影响开始,研究人员就致力于提出各种应对策略来尽量减小这些影响. 这是 2004 年以来人们对 churn 研究的另一个重要研究点.Churn 的影响涉及到 P2P 网络的各个方面,应对 churn 的策略也多种多样,本节从邻居选择策略、失效恢复策略、副本维护策略、连接生命周期维护策略等方面分别对其进行阐述.

5.1　邻居选择策略

邻居选择策略是解决在节点加入网络时如何选择邻居来初始化自己路由表的策略.邻居选择策略不同,应对 churn 的效果就不一样.如果选择的是会话时长非常长的节点,那么 churn 对它的影响将比较小.

Rhea 等人[8]针对非结构化 P2P 网络提出并分

析了一些邻居选择策略:(1)全局抽样:随机抽取前缀为 p 的节点作为节点的邻居;(2)邻居的邻居(Neighbors' Neighbors,NN):选择邻居的邻居作为自己的邻居;(3)邻居的反转邻居(Neighbors' Inverse Neighbors,NIN):选择那些与自己有相同邻居的节点作为邻居;(4)递归抽样(Recursive Sampling):针对邻居的邻居和邻居的反转邻居各自采用递归.文章分别对这些策略进行实验,结果表明,全局抽样的邻居选择策略出人意料的好,NN 和 NIN 并非想象中的那么好,只有加上递归后的 NN 和 NIN 才达到全局抽样的效果.

Yao 等人[22]针对非结构化 P2P 网络提出一种基于年龄的随机邻居选择策略.文章假设节点会话时长服从重尾分布,它将整个 P2P 网络看成一个加权有向图 $G=(V,E)$,对于边 $(u,v)\in E$,$N_u^+=\{v\in V:u\rightarrow v\}$ 表示 u 的出边邻居集,$N_u^-=\{v\in V:v\rightarrow u\}$ 表示 u 的入边邻居集,A_u 表示 u 的年龄,节点 u 的入边权值设置为 u 的年龄和 u 的入边集元素个数的比值:$\omega(v,u)=A_u/|N_u^-|$,u 的入度加权值为 $d_u^-=\sum_{v\in N_u^-}\omega(v,u)=A_u$,出度加权值为 $d_u^+=\sum_{v\in N_u^+}\omega(u,v)=\sum_{v\in N_u^+}(A_v/|N_v^-|)$,则对于基于年龄的随机选择,通过交替地走入边和出边来进行,假设当前节点为 u,则第 1 步选择 u 的一个入边邻居 $h(h\in N_u^-)$,选中的概率为 $p_{uh}=\omega(h,u)/d_u^-$,第 2 步选择 h 的出边邻居 v,选中的概率为 $p_{hv}=\omega(h,v)/d_h^+$,则节点 u 选择 v 作为它的邻居的概率为 $p_{uv}=\sum_{h\in N_u^-}p_{uh}p_{hv}$.

5.2 失效恢复策略

失效恢复策略主要考虑的是在邻居节点失效的情况下,如何进行恢复的策略.

5.2.1 是否立即恢复

根据邻居失效后是否立即恢复可以将其分为反应恢复(Reactive Recovery)和周期恢复(Periodic Recovery).反应恢复是指当节点发现它的邻居表中有节点失效时,立即做出反应,更新其邻居表,并将新的邻居表发给它的每一个邻居.周期恢复是指节点的邻居失效后并不立即进行更新,而是等待一个周期时间 t 后统一进行更新.反应恢复能够及时修复邻居表,维护邻居表的完整性,在低抖动率①时比较高效,但是当抖动率比较高时,节点就会疲于修复邻居表,从而产生较高的维护代价.而周期恢复并不立即进行恢复,而是等待一个周期 t 后,有多少个邻居失效就同时添加多少个邻居,因此,无论在低抖动率还是高抖动率下,周期恢复的性能都比较稳定,从而降低了维护代价.但是高抖动率时,周期 t 的设置比较关键,如果比较长,可能导致节点所有的邻居都失效,而没有及时恢复,从而导致节点被孤立,产生网络分割.Rhea 等人[8]从带宽消耗和查找延时两方面对这两种策略进行比较实验,结果表明,无论在低抖动率还是高抖动率下,周期恢复都比反应恢复要好,而且抖动越明显,周期恢复的优势就越明显.

5.2.2 失效后如何恢复

按照节点失效后是否使用新节点来替代失效节点,可以将策略分为固定(fixed)策略和替代(replacement)策略.固定策略是指不用新节点替代失效节点的策略,替代策略是指节点失效后,立即有新节点来替代该失效节点的策略.进一步,可以将替代策略分为包括随机替代(Random Replacement,RR)策略、被动偏好列表(Passive Preference List)策略以及主动偏好列表(Active Preference List)策略.随机替代策略是指当有节点失效时,从可用节点列表中随机选择一个节点来替代失效的节点.被动偏好列表策略是指将所有可用的节点按照某一属性偏好排序,当邻居表中有节点失效时,从偏好列表中选择最合适的节点来替代失效节点的策略.主动偏好列表策略是指将所有可用节点按照某一属性偏好排序,当邻居表中有节点失效时,从偏好列表中选择最合适的节点来替代失效节点,并且,在使用过程中,发现有比邻居表中更合适的节点上线,则断开已有节点,并切换到该新上线的更合适节点.Godfrey 等人[14]通过一系列的实验和理论分析得出,替代策略比最好的固定策略在长时间的跟踪下有 1.3~5 倍的 churn 降低,当发生节点失效时,简单的随机替代策略的性能比其它策略的性能都要好.偏好列表策略在节点失效的情况下,相比随机替代策略,可能使系统产生更多的抖动.文章最后指出,在实际运用中,适当增加一些随机选择策略可以降低 churn 带来的影响.

5.3 副本维护策略

在 P2P 网络中,churn 可能会导致数据丢失、数据访问延迟增大等问题.为了提高数据可用性,目前大多数 P2P 网络都采用副本技术.但是 churn 同样对副本维护有影响,节点加入可能会产生更好的副

① 抖动率是指单位时间内,网络中节点加入和退出网络的频率.

本存放位置，从而引起副本迁移；节点退出，可能使副本数目减少，从而需要重新选择副本位置，复制副本并将其放到该合适位置上. 副本技术针对 churn 的应对策略主要体现在对副本数目的维护和对副本位置的维护两方面，下面分别对其进行阐述.

5.3.1　对 P2P 副本数目的维护策略

Churn 对 P2P 副本数目的影响主要体现在节点退出可能产生副本丢失，当系统抖动率比较高时，有可能数据的所有副本都丢失而没有及时恢复，从而产生数据丢失. 为了维护数据的可用性及访问的高效性，维护一定数目的副本是必须的，当产生副本丢失时，必须及时恢复. 和邻居失效恢复策略相似，对于副本数目的维护一般有立即恢复和周期恢复两种策略，在低抖动率情况下，立即恢复产生的消息开销比较小，但是当抖动率比较高时，会产生较大的网络开销. 而周期恢复产生的消息开销在两种情况下变化不大，但是在高抖动率情况下，如果周期恢复设定的周期比较长而副本没有及时更新，就有可能产生所有副本均丢失从而导致数据丢失的可能，而立即恢复则不会产生这种情况.

5.3.2　对 P2P 副本位置的维护策略

副本位置的选择对数据副本的稳定性具有较大的影响，当副本存放在短会话时长的节点上时，由于节点退出，导致副本缺失，因此必须重新选择副本节点，如果其仍然是短时长节点，则过一段时间可能又下线了，就会频繁进行“副本缺失——选择副本位置——副本迁移”的过程，从而产生较大的通信开销以及副本维护开销. 但是如果被选中的是会话时长很长的稳定节点，则可以在较长时间内维持副本的存在，降低了副本缺失的概率，减少了网络通信开销和副本维护开销. 因此，一个好的副本位置选择策略对应对 churn 的影响会有更好的效果.

目前来讲，对于非结构化 P2P 网络，由于组织结构比较松散，副本一般是随机分布在网络中，因此并没有特别针对 churn 的相关副本存放策略的研究.

对于结构化 P2P 网络，数据资源(或资源的索引)一般放置在节点 ID 离数据 ID 最近的节点上，称为该数据的根节点(root，Chord 中称为后继 successor). 其副本位置选择一般有两种策略：基于多关键字的选择策略和基于邻居表的选择策略. 基于多关键字的选择策略是指将每个数据资源采用同一个 Hash 函数和不同的 Hash 参数(或者采用不同的 Hash 函数和相同的 Hash 参数)计算得到 s 个不同的 ID 值，每个与 ID 最接近的结点存储数据的一个副本节点. 该策略将根节点从 1 个变成了 s 个，这 s 个根节点只要有一个离开系统，就需要对副本管理信息进行更新，副本也必须迁移. 假设系统中每个节点离开的概率为 p，则发生这种情况的概率为 $1-(1-p)^s$，当 $p=0.5$，$s=5$ 时，其发生的概率为 96.875%. 当系统抖动率比较高时，就可能频繁产生副本缺失，从而产生较高的副本维护开销. 在 Tapestry[31] 和 CAN[32] 中使用的就是该策略.

基于邻居表选择策略是指副本节点在根节点的邻居表中选择，将副本保存在离根节点最近的 s 个邻居中，并由根节点负责管理和维护副本信息. 而 churn 同样对其有影响，当这 s 个邻居中有节点离开网络时，就会产生副本缺失，此时需要将离根节点最近的非副本节点选择为新的副本节点，并将副本复制到该节点. 当有新节点加入并且落在 ID 值离根节点更近的位置时，根据协议此时其成为新的副本节点，而离根节点最远的副本节点就被挤出副本集而成为非副本节点，并需要将该副本迁移到新节点上来，同样产生了副本迁移. 因此，无论是新节点加入还是已有节点退出，都对副本维护产生影响. 在 Chord[33] 和 Pastry[34] 中使用的是该策略.

Legtchenko 等人[35] 在基于邻居表选择策略的基础上提出放宽 DHT(Relax DHT)的副本维护策略. 该策略主要思想是针对副本必须放在离根节点最近的 s 个节点这一限制进行放宽，提出只要是在根节点的邻居表中的节点，都可以存放副本. 而如何在邻居表的众多节点中选择副本节点呢？文章提出了随机副本选择策略，根节点在其邻居表中随机选择 s 个节点用来存放数据的副本. 在数据维护时，新节点加入只要不将副本节点挤出根节点的邻居表，数据就不用迁移，当有副本节点离开网络导致数据副本丢失时，根节点在其邻居表中随机选择一个非副本节点用来存放新的副本. 该策略对新节点加入导致副本迁移具有一定的抑制作用，但是它不能保证随机选择的节点就一定是稳定的节点，虽然经过多次选择后，最终可能会选择到稳定的节点，但是这期间副本可能需要经过多次迁移.

5.4　连接生命周期维护策略

连接生命周期(link-lifetime)是指网络连接从形成到断开所经历的时间. 它和节点会话时长不同的地方在于，连接断开并不意味着节点下线，特别在结构化 P2P 网络中，当有更合适的节点到来时，节点往往主动断开并连接到新节点上，此时，原连接中的两个节点均未下线. 连接生命周期和节点连通性、

网络路由、数据传输都有影响.连接生命周期越长,则节点连通性越强,经过此连接的网络路由就越稳定,数据传输效率也越稳定.

根据节点连接的特点,可以将 P2P 系统分为两类模型:无切换(non-switching)模型和切换(switching)模型[36].无切换模型是指连接一旦建立就一直保持连通直到失效,在连接期间不切换到其它邻居,该模型主要应用于非结构化 P2P 网络中.切换模型是指连接建立以后,如果出现了更合适的连接节点,则主动断开连接并切换连接到新的邻居,该模型主要应用于很多结构化 P2P 网络中.

5.4.1 非结构化 P2P 网络连接生命周期研究

对于非结构化 P2P 网络,Wang 等人[26]针对 Gnutella 网络中的连接生命周期进行测量,由于大多数的 Gnutella 客户端用户都和超级节点相连,因此只要监测超级节点上的连接生命周期就可以获得网络绝大多数的连接情况.通过监测发现,连接生命周期的分布服从幂率(power-law)分布,通过和节点生命周期的对比发现,连接生命周期往往比节点生命周期要短,节点经常更换它们的邻居.监测显示,16.4%的连接在 8 min 后消失.

5.4.2 结构化 P2P 网络连接生命周期研究

对于结构化 P2P 网络,由于一般是切换模型,为了遵循网络的协议规定,旧的邻居经常被新到来的邻居替换,而新邻居又往往具有非常短的剩余时长,从而给网络带来较高的连接抖动率.Tan 等人[37]提出一种在线最长邻居选择策略(Longer-Lived Neighbor Selection, LNS).它首先假设网络中节点的会话时长服从 Pareto 分布,在此假设下,在线时长越长的节点,其剩余时长也越长.因此,选择在线时长越长的节点,其连接生命周期也越长.为此,文章提出如下的选择策略:节点 V 的第 i 个邻居应在 $[V+2^i, V+2^{i+1})$ 之间,设此区域中所有节点组成的集合为 S_i,当 i 较小时,S_i 中的元素非常少,当 i 增加时,S_i 可能非常大,此时从 $[V+2^i, V+2^{i+1})$ 中随机选择最多不超过 K(K 为某一设定值)个节点,组成集合 S_i.然后从 S_i 中选择在线时长最长的节点作为节点 V 的第 i 个邻居.该方法的优点是,在线时长最长的节点往往具有较长的稳定性,因此连接比较稳定,减小了 churn 对 DHT 的影响.

Yao 等人[36]在其基础上提出一种小区域(min-zone)邻居选择策略.节点 V 的区域是指从节点 V 到它的后继之间的一小部分 DHT 空间,由 V 负责.和文献[37]一样,文章假设到节点 V 的距离为 $2^i \sim 2^{i+1}$ 之间的节点集合为 S_i,它首先在 S_i 中平均随机地选择 m 个节点,然后将 V 连接到这 m 个节点中所负责区域最小的那个节点上.该方法的优点是:由于负责的区域较小,新加入的节点落在该区域的概率较小,降低了新加入节点对其的影响.但是小区域方法虽然可以降低新节点加入的影响,但不能降低旧节点离开的影响,有些新加入的节点切割原有区域从而使自己成为小区域节点,当连接切换到这些节点后,这些节点可能具有较短的会话时长,没过多久就离开网络,从而产生抖动.如何降低节点离开对 DHT 的影响,是小区域方法需要改进的地方.

5.5 针对具体问题的应对策略研究

由于 churn 对 P2P 网络的影响涉及到各个方面,针对不同的问题,研究人员提出了有针对性的应对策略.

5.5.1 查询超时设置策略

Rhea 等人[8]针对 churn 对网络查询的影响进行研究.在 churn 下,简单设定查询应答的超时值是不明智的,如果查询超时设定太短,查询请求可能刚被应答节点接收到、或者正在处理中、或者查询应答在返回的途中.如果查询超时设定值太长,则请求节点往往无法忍受等待太长时间而离开网络.文章针对 churn 对超时设定的影响,提出 3 种超时设定的策略:(1)固定为 5 s;(2)TCP-style 超时:根据以往的反应时间来设定现在的超时值;(3)虚拟纵轴:根据网络中两节点距离来设定超时值.并对这 3 种策略进行实验比较,结果表明,当抖动率比较低时,3 种策略的差别不大,但是当网络中节点的平均会话时长小于 23 min 时,TCP-style 超时设定的优势比较明显.

5.5.2 系统稳定性维护策略

Churn 对 P2P 系统的稳定性产生重要影响,受传统分布式系统的启发,Kuhn 等人[38]提出建立一个动态的 P2P 系统来应对 churn,该系统的基本结构是一个扁平图,图中的每一个节点(node)都是由 $O(d^2)$ 个终端节点(peer)组成,其中 d 表示 node 的边长,在每个 node 内部的 peer 都相互连接起来,相邻的 node 之间也通过一些 peer 连接起来.在抖动情况下,一些 peer 可能会下线或失效,此时 node 中 peer 的数目就会不一样,文章通过一系列的算法来选择数目比较多的 node 中的 peer 转移到数目较少的 node 中,来确保系统中的每个 node 都具有大致相同数量的 peer.当 peer 加入系统时,也通过相应的查找和定位算法,来找到 peer 数目较少的 node,

并加入到该 node 中. 当 peer 的总数增加或减少到一定的阈值后，系统的阶数(order，指系统中 node 的数量)就相应地增加或减少 1. 该系统的优点是，网络中 node 的数目受 churn 的影响比较小，node 之间的连接能够长期保持，从而使整个系统系统保持稳定.

6　总结与展望

P2P 应用是目前互联网上最流行的应用之一，提高 P2P 系统的性能具有很重要的理论价值和应用价值. 但是抖动特性严重影响了 P2P 的性能、可靠性，制约了 P2P 的推广应用.

抖动的研究目前还是热点问题，从目前的发展趋势看，它主要朝以下几个方向进行：

(1) 针对实际运行的结构化 P2P 网络进行研究是目前的热点之一. 目前，基于 Kademlia 网络的 P2P 应用软件 eMule、BitTorrent 等在互联网上大量流行，拥有非常庞大的用户群体. 在这些软件上进行抖动特性的研究，进而根据研究结果对这些软件进行相应的改进，既有理论价值，又和实际应用相结合. 目前 eMule 中存在以下几点值得深入挖掘和进一步研究.

提高网络连接的稳定性. 这体现在两个方面：① 连接的节点时常被断开. 在 KAD 网络的连接显示中，有些已经连接上的节点，过一段时间去查看时，就显示已经断开了. ② 数据传输速率变化非常大. 在数据传输过程中，传输速率在较短的时间内从上百 KB/s 减小到几 KB/s，然后又增加到上百 KB/s，这种过山车一样的传输速率变化是网络连接不稳定的最直观体现. 如何针对 KAD 网络的动态性进行节点的选择，使节点的连接在一个相对较长的时间内保持稳定，同时使数据传输速率能够像迅雷一样，在一个较长时间内保持相对稳定. 是目前切实存在而又亟需解决的研究热点之一.

及时查找出不存在网络中的节点. 在 KAD 网络中，有些节点虽然表面显示存在网络中，但是实际连接时，却并不存在. 这些表面存在而实际失效的节点，增加了网络在资源搜索、消息路由、数据传输过程中的代价，降低了系统的性能. 如何及时找出这些表面存在实际却不在网络中的节点，并通知其它节点及时删除这些节点的信息，从而避免无谓的连接，提高资源搜索、消息路由等的性能. 是在 KAD 网络中可以进一步研究的另一个研究点.

(2) 在移动 P2P 网络中的 churn 特性研究是目前研究的另一个热点. 随着移动互联网的兴起，移动 P2P 网络也热起来. 和传统 P2P 一样，移动 P2P 网络中自然也具有 churn 特性，而且比传统网络中的更严重. 由于移动设备具有位置经常变化，需要节约能耗等方面的因素，节点加入和退出网络更频繁. 同时，它还有不同于传统 P2P 网络的自身特点，传统 P2P 网络中，节点位置一般比较固定，该次上线时地理位置比较近的邻居节点，下次上线时，往往还是地理位置比较近的节点，因此选择这些节点，数据的上传下载效率都比较高. 而移动 P2P 网络却不一样，由于位置的变化，该次上线时距离较近的节点，下次可能相隔就很远了，如果像上次上线时一样，选择这些节点，那么连接的稳定性、网络传输效率可能就会很低. 针对这些问题探讨其可能的解决方案是 churn 研究的另一个发展方向.

致　谢　衷心感谢评审专家付出的辛勤劳动和对本文提出的中肯意见. 感谢课题组的老师和同学们提出的宝贵意见，特别感谢丁博、杨永志师兄，文章的多次修改都离不开他们提出的宝贵意见！

参　考　文　献

[1] Pouwelse J A, Garbacki P, Epema D H J et al. The BitTorrent P2P File Sharing System: Measurements and Analysis//Proceedings of the IPTPS'05. New York, 2005: 205-216

[2] Zheng Wei-Min, Yu Hong-Liang, Shi Guang-Yu et al. Parallel discrete event simulation based large-scale P2P system behavior pediction. Science China: Information Sciences, 2010, 40(10): 1338-1350(in Chinese)
(郑纬民，余宏亮，施广宇等. 基于并行离散事件模拟的大规模 P2P 系统行为预测. 中国科学：信息科学，2010，40(10)：1338-1350)

[3] Shirky C. Listening to Napster. Oram A ed. Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology, O'Reilly, 2001

[4] Ripeanu M. Peer-to-peer architecture case study: Gnutella network//Proceedings of the International Conference on Peer-to-Peer Computing. Sweden, 2001: 99-100

[5] Bhagwan R, Savage S, Voelker G M. Understanding availability//Proceedings of the IPTPS'03. California, 2003: 256-267

[6] Chu J, Labonte K, Levine B N. Availability and locality measurements of peer-to-peer file systems//Proceedings of the ITCom'02, Boston, MA, 2002

[7] Saroiu S, Gummadi P K, Gribble S D. A measurement study of peer-to-peer file sharing systems//Proceedings of the

MMCN. San Jose, CA, 2002

[8] Rhea S, Geels D, Roscoe T et al. Handling churn in a DHT. UC Berkeley: Computer Science Technical Report UCB/CSD-3-1299, 2003

[9] Rhea S, Geels D, Roscoe T et al. Handling Churn in a DHT//Proceedings of the USENIX Annual Technical Conference. Boston, 2004: 127-140

[10] Maymounkov P, Mazieres D. Kademlia: A peer-to-peer information system based on the XOR metric//Proceedings of the IPTPS'02. Cambridge, 2002: 53-65

[11] Zhang Yu-Xiang, Yang Dong, Zhang Hong-Ke. Research on Churn problem in P2P networks. Journal of Software, 2009, 20(5): 1362-1376(in Chinese)
(张宇翔, 杨冬, 张宏科. P2P 网络中 Churn 问题研究. 软件学报, 2009, 20(5): 1362-1376)

[12] Lindley N H, Churn Perry W. United States Patent Office, 1843(2993): 1-2

[13] Li J, Stribling J, Gil T M et al. Comparing the performance of distributed hash tables under churn//Proceedings of the IPTPS'04. California, 2004: 87-99

[14] Godfrey P B, Shenker S, Stoica I. Minimizing churn in distributed systems//Proceedings of the ACM SIGCOMM. New York: ACM Press, 2006: 147-158

[15] Stutzbach D, Rejaie R. Understanding churn in peer-to-peer networks//Proceedings of the 6th ACM SIGCOMM on IMC. New York: ACM Press, 2006: 189-202

[16] Ripeanu M. Peer-to-peer architecture case study: Gnutella network//Proceedings of the IEEE P2P'01. Sweden, 2001: 99-100

[17] Gummadi K P, Dunn R J, Sariou S et al. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload//Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP 2003). New York: ACM Press, 2003: 314-329

[18] Bustamante F E, Qiao Y. Friendships that last: Peer lifespan and its role in P2P protocols//Proceedings of the 8th International Workshop on Web Content Caching and Distribution (WCW 2003). Norwell: Kluwer Academic, 2003: 233-246

[19] Stutzbach D, Rejaie R. Characterizing churn in peer-to-peer networks. University of Oregon, Oregon: Technical Report CIS-TR-2005-03, 2005

[20] Steiner M, En-Najjary T, Biersack E W. Long term study of peer behavior in the KAD DHT. IEEE/ACM Transactions on Networking, 2009, 17(6): 1371-1384

[21] Saroiu S, Gummadi K P, Gribble S D. Measuring and analyzing the characteristics of Napster and Gnutella hosts. Multimedia Systems, 2003, 9(2): 170-184

[22] Yao Z, Wang X, Leonard D et al. Node isolation model and age-based neighbor selection in unstructured P2P networks. IEEE/ACM Transactions on Networking, 2009, 17(1): 144-157

[23] Memon G, Rejaie R, Guo Y et al. Large-scale monitoring of DHT traffic//Proceedings of the IPTPS'09. Boston, 2009

[24] Stutzbach D, Rejaie R. Characterization of P2P systems. Handbook of Peer-to-Peer Networking, Springer, 2009

[25] Sen S, Wang J. Analyzing peer-to-peer traffic across large networks. IEEE/ACM Transactions on Networking, 2004, 12(2): 219-232

[26] Wang X, Yao Z, Loguinov D. Residual-based measurement of peer and link lifetimes in Gnutella networks//Proceedings of the IEEE INFOCOM. Alaska, 2007: 391-399

[27] Steiner M, Biersack E W, En-Najjary T. Actively monitoring peers in KAD//Proceedings of the IPTPS'07. Washington, 2007: 26-27

[28] Zhou Mo, Zhang Jian-Yu, Dai Ya-Fei. Design and optimization of the scalable DHT Web crawler. Science China: Information Sciences, 2010, 40(9): 1211-1222(in Chinese)
(周模, 张建宇, 代亚非. 可扩展的 DHT 网络爬虫设计和优化. 中国科学: 信息科学, 2010, 40(9): 1211-1222)

[29] Stutzbach D, Rejaie R, Duffield N et al. On unbiased sampling for unstructured peer-to-peer networks. IEEE/ACM Transactions on Networking, 2008, 17(2): 377-390

[30] Rasti A H, Torkjazi M, Rejaie R et al. Respondent-driven sampling for characterizing unstructured overlays//Proceedings of the IEEE INFOCOM-Mini. Rio de Janeiro, 2009: 2701-2705

[31] Zhao B, Kubiatowicz J, Joseph A A. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Computer Science Division. University of California, Berkeley: Technical Report UCB/CSD-01-1141, 2001

[32] Ratnasamy S, Francis P, Handley M et al. A scalable content-addressable network//Proceedings of the SIGCOMM'01. San Diego, CA, 2001: 161-172

[33] Stoica I, Morris R, Karger D et al. Chord: A scalable peer-to peer lookup service for Internet applications//Proceedings of the SIGCOMM'01. San Diego, 2001: 149-160

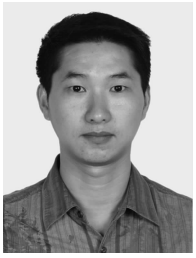
[34] Rowstron A, Druschel P. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems//Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001). Heidelberg, 2001: 329-350

[35] Legtchenko S, Monnet S, Sens P et al. Churn-resilient replication strategy for peer-to-peer distributed hash-tables//Proceedings of the 11th International Symposium on Stabilization, Safety, and Security of Distributed Systems. Lyon, 2009: 485-499

[36] Yao Z, Loguinov D. Link lifetimes and randomized neighbor selection in DHTs//Proceedings of the IEEE INFOCOM. Phoenix, Anchorage, 2008: 146-150

[37] Tan G, Jarvis S A. Stochastic analysis and improvement of the reliability of DHT-based multicast//Proceedings of the IEEE INFOCOM. Anchorage, AK, 2007: 2198-2206

[38] Kuhn F, Schmid S, Smit J et al. A blueprint for constructing peer-to-peer systems robust to dynamic worst-case joins and leaves//Proceedings of the IWQOS'06. New Haven, 2006: 12-19



FU Zhi-Peng, born in 1981, Ph. D. candidate. His research interests include P2P network, distributed computing.

WANG Huai-Min, born in 1962, Ph. D., professor, Ph. D. supervisor. His research interests include distributed computing, information security and computer software.

SHI Dian-Xi, born in 1966, Ph. D., professor. His research interests include distributed computing, middleware, pervasive computing.

ZOU Peng, born in 1957, Ph. D., professor, Ph. D. supervisor. His research interests include distributed computing, operating system.

Background

P2P application is one of the most popular applications in the Internet, but its spread is limited by the churn character of the network. The frequent arrival and departure of thourands or millions of nodes is badly influent the function, performance and the reliability of the P2P network. Studying churn and presenting a series of useful strategies to resilient the churn have theoretical value and application value.

In this paper, we reviews the origin of the churn, the definition of it and its influence to the P2P network, and minutely describe the churn from the profile of the statistical properties, measurement methods and resilient strategies. We hope that through this way we can help the researchers to card the development trace of the churn, and to know the future trend of it. This work is supported by the National Natural Science Foundation of China under Grant No. 90818028: "Behavior Monitoring and Trustworthiness-Oriented Evolution of Large-Scale Distributed Software Systems" and National Grand Fundamental Research 973 Program of China under Grant No. 2011CB302600: "Basic Research of Effective and Trustworthy Internet-Based Virtual Computing Environment(iVCE)".