软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(6):1301−1315 [doi: 10.13328/j.cnki.jos.004435]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

结构化对等网测量方法研究*

闫 佳[1], 应凌云[1], 刘海峰[2], 苏璞睿[1], 冯登国[1]

[1](中国科学院 软件研究所,北京 100190)

[2](北京信息安全测评中心,北京 100101)

通讯作者: 闫佳, E-mail: yanj@is.iscas.ac.cn

摘 要: 网络测量是深入开展结构化对等网研究的基础,结构化对等网络协议设计、共享内容检索、态势感知乃至安全性的研究都需要以网络测量为前提.在节点分布对等、实时变化显著、未知瞬发扰动频繁的结构化对等网络中,获得其准确、完整的网络信息更是十分困难的.通过形式化分析结构化对等网节点搜索过程,研究节点信息在全网分布情况与查询返回率之间的关系,将历史测量数据与具体对等网特征信息相结合挖掘节点搜索优化策略,提出了一种网络资源占用显著降低、搜索速度较快、信息完备率较高的搜索测量优化方法.KAD 网络是目前得到大规模部署运行的为数不多的结构化对等网络之一,以 KAD 网络为主要研究对象开发了 KadCrawler 对等网搜索系统,进行了大量测量和分析,验证了搜索优化方法的可行性和有效性;同时,对当前 KAD 网络拓扑结构特征、节点重名等现象进行了初步分析,发现 KAD 网络近年来发生了显著的变化.

关键词: 网络测量;P2P;结构化对等网络;Kademlia;KAD 网络

中图法分类号: TP393

中文引用格式: 闫佳,应凌云,刘海峰,苏璞睿,冯登国.结构化对等网测量方法研究.软件学报,2014,25(6):1301−1315 http://www.jos.org.cn/1000-9825/4435.htm

英文引用格式: Yan J, Ying LY, Liu HF, Su PR, Feng DG. Research on network measurement of structured P2P network. Ruan Jian Xue Bao/Journal of Software, 2014,25(6):1301−1315 (in Chinese). http://www.jos.org.cn/1000-9825/4435.htm

Research on Network Measurement of Structured P2P Network

YAN Jia[1], YING Ling-Yun[1], LIU Hai-Feng[2], SU Pu-Rui[1], FENG Deng-Guo[1]

[1](Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

[2](Beijing Information Security Test and Evaluation Center, Beijing 100101, China)

Corresponding author: YAN Jia, E-mail: yanj@is.iscas.ac.cn

Abstract: Network measurement is the foundation for the in-depth research on P2P network. It's a prerequisite for the P2P protocol design, shared content searching, situational awareness as well as research on the security of P2P network. In structured P2P network with decentralized peer to peer relationship, high dynamics and unpredictable instantaneous disturbance, achieving highly accurate and near-complete information retrieval is much more difficult. This paper formalizes the search (or crawl) process of structured P2P network, studies the relationship between the node's route spreadness in the whole network and the query response rate in the midst of crawling, derives some improved search strategies from the knowledge of historic measurements and characteristics of specific P2P network, and proposes a improved search method with much lower bandwidth consumption, fast crawling speed as well as relatively high coverage of nodes in structured peer-to-peer network. KAD network is among the few of structured P2P networks that are extensively deployed. This research mainly concentrates on KAD network and develops a search tool called KadCrawler, upon which large amounts of measurements and analysis are conducted. The result shows that the proposed method is both feasible and effective. Lastly, an analysis on the topology and phenomenon of ID repetition reveals that KAD network has changed significantly over the years.

* 基金项目: 国家自然科学基金(91118006, 61073179); 国家高技术研究发展计划(863)(2011AA01A203); 国家重点基础研究发展计划(973)(2012CB315804); 北京市自然科学基金(4122086)

收稿时间: 2012-09-14; 定稿时间: 2013-06-09

Key words: network measurement; P2P; structured P2P network; Kademlia; KAD network

对等网络在文件共享、流媒体等领域得到广泛应用,实时测度其网络特征十分必要,因而对等网络的测量一直是学术界研究的重点.设计更好的结构化对等网络,以达到高效率、低资源消耗、高可靠健壮性的目标,也必须能够实时追踪网络节点整体分布信息,实现对等网络的有效测量.要达到可测量这一目标,必须首先对节点信息的快速搜索方法进行研究.结构化对等网络的节点信息搜索可以得到网络中大量活跃节点的基本信息,这些信息可以有效反映网络整体在某个时间点或时间段内的真实状况.

当前的节点信息搜索测量主要面临下面 3 个难点:

(1) 节点信息搜索普遍采用的全遍历搜索需要查询每个已知的节点,在带宽消耗巨大的同时,效率和速度都较低.如 Steiner 等人使用的 Blizzard[1]、Liu 设计的 Rainbow[2]都存在这个问题;

(2) 缺乏可供参照验证的基础数据和评测标准.研究者往往基于各自的测量结果进行分析,得出不同结论,难以进行比较和评价;

(3) 网络扰动给测量带来的误差难以消除.测量时间过长导致较大随机误差,测量方法差异会导致不同的系统误差[3].

针对上述难点,本文在总结前人搜索算法的基础上分析搜索过程,提出 5 项测评指标,分别从测量效率、测量代价等角度对搜索方法进行评价,为不同方法的比较奠定基础.依据分析结果,我们提出了一种优化搜索方法,基于前述 5 项测评指标的测量实验结果表明,该方法占用资源更小、效率更高.使用该搜索方法,我们针对 KAD 网络进行测量,发现其规模缩小并且其中非活跃节点占到了 90%以上.

本文第 1 节介绍对等网搜索测量的相关工作,包括被动、主动的测量方式,以及其他研究工作中不同测量结果的对比.第 2 节针对结构化对等网的搜索过程进行形式化分析,给出节点路由延展指数以及查询后继活跃率等指标的概念和定义,通过分析这两个指标对搜索性能指标的影响,提出一种有效减少测量带宽消耗和搜索测量时间的优化方法.第 3 节使用该优化方法针对 KAD 网络进行搜索测量以及测量效果评估.第 4 节给出结论和展望.

1 相关工作

对等网络测量指在对等网络的网络层及其以上数据处理栈上进行的微观信息获取、宏观特征分析以及网络整体态势的度量和评估.测量范畴主要包括获取对等网节点信息、节点间关系等相对静态特征以及网络宏观运行时特征等.

网络测量的基础是对网络中信息进行有效而充分地检索,按照获取的方式可以简单地分为被动式和主动式两种.

被动式测量指在对等网络中部署监听节点或者选定一些节点进行带外数据监听,追踪单个节点或者少量节点在某个连续时间段内的行为[4,5].这种方式操作简单,占用资源相对较少,对系统和网络的运行干扰很小,同时可以提供丰富的细节信息,能够在时间维度上对行为进行统计分析以形成对网络中一般节点行为的预期和估计,为网络态势评估提供一定依据.被动式测量的缺点是获取宏观统计分析所需的大量样本数据比较困难,难以得到统计意义上显著的分析结果.

主动测量主要是指从单个或多个网络节点出发,遵循协议规范要求向网络中其他节点发送查询,以获得其他节点的信息.主动测量更类似于一次网络搜索过程,因此,本文中搜索与测量可以互换使用.与被动式测量方法相比,基于主动查询的信息获取方式不受限于局部网络的有限流量数据,可以获得更大范围内节点的有关信息.如果搜索速度足够快的话,短时间内获得的网络节点群可以近似作为网络整体的一个快照,从中可以分析得到对等网的一些全局特征.如:Stutzbach 等人利用 cruiser[3]系统不仅能够对非结构化对等网络 Gnutella 进行测量,并对 KAD 等结构化对等网进行了基于部分固定区域的搜索;王勇等人[6]使用主动探测技术测量了非结构化的 Gnutella 网络,获得了大量搜索数据;Steiner[1]利用高带宽资源,使用主动快速抓取方法得到节点信息镜像.

结构化对等网络通信去中心化的特点,使得整个网络在通信模式上呈现高度对称的平面化结构.由于文件共享是得到大规模部署的结构化对等网络最常见的应用模式,而大多数应用都混合了半结构化对等网以提高效率,因此其整体结构中引入的不对等性,会对测量造成一些干扰.如大多数 P2P 下载软件均部分集成了结构化对等网以提高可用性,多种对等网会交叉在一起共同组网,因此在测量过程中必须对这种情况予以区分.我们把问题限定在通信对等的范围内,即暂时不考虑节点同时参与的多个对等网之间的相互影响,而只关注其在指定的结构化对等网中所实施的行为.对于被动测量来说,需要能够区分多种对等网不同的流量特征.对于主动测量来说,则需要设计实现独立的对等网协议交互工具.

Kademlia 协议[7]是目前少数得到大规模部署的对等网协议之一,eMule 软件内置的 KAD 文件共享网络,是其中最具代表性的实现之一,因此,本文选取 KAD 网络作为我们测量研究的对象.

由于 KAD 网络缺少有效的身份验证机制和会话机制,任何一个节点只要遵循 KAD 协议的要求就可以接入网络获取路由信息、共享文件信息、发布文件信息,并且各个消息之间协议约定的关联性很弱.这样的设计可以提高网络承担不同负载时的伸缩性,增强协议扩展的灵活性,但在降低网络接入门槛的同时,使得剔除虚假节点的难度增高,估计节点数量和分析网络宏观特征十分困难.

结构化对等网络拥有与非结构化对等网络完全不同的网络拓扑特征,王勇等人[6]对非结构化对等网络的搜索方法进行了比较.这里,我们针对目前结构化对等网测量方法的研究成果,按照搜索速度、搜索节点数量、搜索所需的带宽、主机等资源消耗进行了分析比较,结果见表 1.其中,NA 代表没有公开该项数据,搜索范围中的 zone 指结构化对等网协议约定依据某个固定的算法生成的数字 ID 进行缩小划分而形成的区域.由于 ID 生成算法一般有较好的随机性,因此认为固定区域中的节点与全网节点在某些特征上是一致的,Chord,CAN, Pastry 等结构化对等网[8−10]也具有类似的性质.

Table 1 Comparison of different structured P2P network search tools

表 1 结构化对等网搜索工具性能对比

搜索方法	速度(分钟)	节点数量(万)	消耗带宽(packets)	主机数量	支持协议	搜索范围
Cruiser[3]	2~8	1~2	NA	1	KAD,BT DHT	zone
Blizzard[1]	8	300~430	NA	1	KAD	zone, full
Rainbow[2,11]	NA	NA	NA	1	KAD	full
周模[12]	90	3	NA	1	KAD	zone
QiWu[13]	4	2 153 775	NA	分布式	KAD	full, zone
JieYu[14,15]	25~40	250~300	NA	1	KAD	full

由表 1 可看出,在给出具体数据的网络信息搜索工具中,大多数都以区域搜索为主.以全局搜索为主的少数几种工具则需要花费较长的时间或者依赖于高性能的硬件配置和高带宽的网络环境.Wu 等人[13]提出的分布式搜索具备较强的性能优势,但是各个节点之间数据共享的传输效率和信息协同处理仍然是一个制约性能的瓶颈.此外我们注意到:表 1 的各种测量方法均没有考虑搜索代价问题(消耗带宽),多数方法对于主机硬件和网络性能的要求较高.因此,本文通过分析搜索过程中的带宽消耗提出一种优化方法,显著降低搜索代价的同时提高了搜索速度,可以在低性能和低带宽的主机上实施有效测量.

2 结构化对等网络搜索的分析与优化

本节首先简要介绍结构化对等网的组成特点以及被广泛采用的朴素搜索查询算法,然后对搜索过程进行形式化建模与分析,并在此基础上提出一种主动测量的优化方法.

2.1 结构化对等网结构特点

结构化对等网通常包含如下 4 类机制:

1) 节点身份生成与识别,加入网络和启动自身的机制.常用的消息原语有启动(bootstrap)等;
2) 节点之间路由交换和更新维护机制.常用的消息原语有存活检查(hello request)、路由查询(route query)等;

3) 支持上层应用的其他机制.比如键值对(key-value)存储机制,常用的消息原语有键值插入和删除等(put key&value,delete key&value);

4) 节点退出以及相应的网络自调整机制.常用的消息原语有路由更新通知(route BroadCast).

可以看到:除了第 3 个机制外,都可以作为从一个节点获取其他更多节点信息的手段,这为结构化对等网的信息搜索提供了多种可能的选择.我们在讨论结构化对等网搜索方法的设计原则时,对可能的信息获取方法进行了比较和优选,详见第 3.1 节.本文中,一般测量搜索方法的设计暂不考虑多种具体搜索方式可能带来的影响.

2.2 结构化对等网的朴素搜索算法

朴素的搜索查询算法[1,3,14,15]采用广度优先策略,使用第 2.1 节提到的第 2 类消息类型来查询网络中的节点.该方法被广泛采用,由查询发送线程和应答接受线程组成,两个线程共享一个待发送节点队列:查询发送线程从该队列获取目标节点;应答接受线程接受消息反馈,解析得到新的节点信息并依次添加到待发送节点队列中.基于上述朴素搜索算法,本文提出了对应的优化方法.

2.3 结构化对等网搜索过程形式化分析

本节对结构化对等网搜索过程中的关键步骤和有关数据参数进行了形式化描述和建模分析,提出了一种网络搜索优化方法以及 5 个搜索器性能评价指标.

2.3.1 搜索过程中重要参数和步骤的形式化描述

假设时刻 t 进行的一次搜索能够获得的节点信息(包括初始的种子节点)的集合定义为 NS_{all}(本文的形式化描述和定义如果依赖不同的时间参数则必须用不同的 t 参数来表示时间概念,否则默认表示处于相同时刻,隐去时间参数 t),该集合中没有重复节点,其数量为$|NS_{all}|$,本文在后面用带下标的 N,如 N_i,或不带下标的 N 来表示任意一个搜索到的普通节点.

结构化对等网搜索的基本方法是通过不断地向已知节点发出查询来获得更多其他节点的信息,迭代查询的过程中有两个步骤十分关键:

1) 是否查询新发现的节点;

2) 对于每个作为查询目标的节点,发送什么样的查询消息.

针对步骤 1),本文定义节点选择布尔函数$\Lambda(N_i,i=1,2,\ldots,s)$,表示搜索器是否选中该节点作为查询目标:

$$\Lambda(N_i,i=1,2,...,|NS_{all}|)=\begin{cases}1, & \text{选作查询目标}\\ 0, & \text{不选作查询目标}\end{cases}.$$

搜索器能够获取的信息全部来自于被选择作为查询目标的节点,定义这些节点的集合为 SP_{all_send},其占全部$|NS_{all}|$个节点的比例为 P_{send}.那么有以下等式成立:

$$|SP_{all\_send}|=|NS_{all}|\bullet P_{send}=\sum_{N\in NS_{all}}\Lambda(N).$$

针对步骤 2),本文定义节点查询消息函数$\Phi(N)$,表示搜索器向节点 N 发送的消息集合.该函数定义了向节点 N 发送消息的数量和内容,则搜索器采用节点查询消息函数Φ在一次搜索中所需要发送的消息总数为

$$\omega(\Phi)=\sum_{N\in SP_{all\_send}}|\Phi(N)|.$$

在清晰描述搜索中两个重要步骤后,定义从时刻 t 开始的、采用节点选择布尔函数Λ和节点查询消息函数Φ的一次搜索 $Crawl_t(\Lambda,\Phi)$耗时为 $T_t(\Lambda,\Phi)$,搜索到的节点数量为 $Size_t(\Lambda,\Phi)$.那么,从时刻 t 开始的一次朴素搜索查询 $Crawl_t(\Lambda_{native},\Phi_{native})$的搜索耗时为 $T_t(\Lambda_{native},\Phi_{native})$,搜索到的非重复节点数量为 $Size_t(\Lambda_{native},\Phi_{native})$.

2.3.2 搜索性能评价指标

提高搜索器的测量性能有两种方法,即提升搜索器所利用硬件设备的性能、网络带宽以及搜索算法的优化.本文主要讨论后者,限定硬件条件下的搜索算法性能优化,重点关注如何降低搜索过程中发送的消息数量,以提高搜索速度和减少搜索时间.通过前一节的分析可见,达到这一目标有两条途径:首先是选择恰当的节点选择布尔函数Λ以减少作为查询目标的节点数量$|SP_{all_send}|$;其次是选择恰当的节点查询信息函数Φ,在确保搜索结

果完备性的同时减少每个节点所需要耗费的查询消息数量$\omega(\Phi)$.因此,有必要对搜索的效率和结果完备性等需求进行分析,解决如何对搜索方法的性能进行评价和比较的问题.

首先关注在搜索结束后能够获得的信息,搜索中所有响应查询的节点信息的集合定义为 C_{alive},并且定义其对应的节点活跃函数为 C_ALIVE.目标节点中响应回复的节点比例为 P_{send_alive},称其为查询活跃率.以上概念可以形式化地表示(其中,$\sigma(N\in SP_{all_send})$为任意一个节点 N 响应搜索器的查询而返回的其他节点的集合):

$$\begin{cases} C_{alive}=\{N_i:\text{if}\mid\sigma(N_i)\neq 0\mid\} \\ P_{send\_alive}=\dfrac{\mid C_{alive}\mid}{\mid SP_{all\_send}\mid} \\ C\_ALIVE(N\in SP_{all\_send})=\begin{cases}1, & N\in C_{alive}\\ 0, & N\notin C_{alive}\end{cases}\end{cases}.$$

假设从时刻 t 开始的时间段 T 内(搜索器查询需要一段时间,在这段时间内的扰动是不可忽略的[1,3],而且查询结果在不同的时间也显著不同),在线节点的数量为 S_{alive},其中,响应搜索器查询的节点数量所占比例记为 P_{alive},即搜索器在从时刻 t 开始的时间段 T 内能够找到的节点中活跃节点的数量为 $S_{alive}P_{alive}$,用 SP_{alive} 来表示,同时,我们有 $C_{alive}=SP_{alive}$.作为搜索器的能力指标之一的 P_{alive},获得准确数据甚至去估测都是比较困难的,一些实测中的 P_{alive} 甚至不到 50%,已经有学者[1,3,14]对这一现象背后的原因进行了初步讨论.而查询活跃率 P_{send_alive} 易于计算,可以反映不同搜索器的性能,因此,本文采用 P_{send_alive} 代替 S_{alive} 作为一项搜索器性能评价指标.

其次,由于结构化对等网固有的特点,不同节点返回的其他节点信息存在大量冗余,即搜索结果中的一个节点可能包含在多个其他节点返回的查询应答消息中.该现象可以形式化地表示为

$$(N_a\in NS_{all})\Rightarrow\exists N_b((N_b\in NS_{all})\wedge(N_a\in\sigma(N_b))\wedge(N_b\neq N_a)),$$

其中,有 $NS_{all}=\bigcup_{N\in SP_{all\_send}}\sigma(N)$ 成立.为了在整体上描述搜索结果中的冗余现象,籍以定量地对不同搜索方法进行比较,这里引入搜索冗余指数 SRI(search redundancy index),定义为返回的结果中去除冗余的节点数量占全部节点数量的比例,即:

$$SRI=\frac{\mid NS_{all}\mid}{\sum_{N\in SP_{all\_send}}\mid\sigma(N)\mid}=\frac{\left|\bigcup_{N\in SP_{all\_send}}\sigma(N)\right|}{\sum_{N\in SP_{all\_send}}\mid\sigma(N)\mid}.$$

为了综合评价不同搜索算法的优劣,本文采用搜索时间 $T_t(\Lambda,\Phi)$(时刻 t 开始搜索直到结束时的时间段)、搜索消耗的带宽资源(即发送的查询消息数量$\omega(\Phi)$)、搜索到的无重复节点数量$|NS_{all}|$、查询活跃率 P_{send_alive}、搜索冗余指数 SRI 来作为衡量搜索性能的指标.这 5 种性能指标很难同时达到最优,不同的测量需求对应不同的性能指标.如:获取全网节点集合的快速镜像时,为了缩短搜索时间、提高搜索结果中节点数量,那么就需要消耗较多的带宽资源,这样 SRI 指数会较小;当需要快速获取网络中活跃节点的集合镜像时,就要提高查询活跃率,最终获取的所有节点数量就会偏小.

本文的搜索优化方法重点关注如何减少查询带宽消耗、缩短查询时间,因此主要是对Λ函数和Φ函数进行优化.

2.3.3 搜索过程中节点选择Λ函数的优化

选择恰当的节点选择布尔函数Λ可以有效提高 P_{send_alive},即提高目标查询节点的应答回复率.由于结构化对等网中节点分布广泛,网络具有高度动态变化性,导致任意时刻都有一定数量的节点加入和退出,搜索器在一个时间段内获得的节点中有一部分为离线节点,这部分节点与该时段内活跃的节点是很难区分的,因此构造能够有效提高搜索效率的Λ函数是比较困难的.为了找到优化Λ函数的方法,下面本文会探讨搜索过程中的两个相关概念以及它们的一些特征.

测量中除了初始启动节点以外,所有节点信息都是从其他节点的查询应答中获取的.为了描述这种关系,定义 $R(N)$为一次完整的搜索过程中从节点 N 获取到的所有其他节点的集合,称作关于节点 N 的直接搜索后继,那么本文定义如下函数来表示节点 N_i 是否在节点 N_j 返回的查询应答中:

$$InRouteOf(N_i,N_j)=\begin{cases}1, & N_i\in R(N_j)\\ 0, & N_i\notin R(N_j)\vee i=j\end{cases}.$$

结构化对等网节点路由维护的特点决定了一个节点的信息可能出现在多个其他不同节点的路由表中,即可能出现在多个不同查询目标节点的应答消息中.为此,本文引入两个概念来描述节点的这一动态属性:路由延展指数(route stretchness indicator,简称 RSI)和路由搜索延展指数(route search stretchness indicator,简称 RSSI).一个节点 N 的 RSI 定义为某个搜索过程中,在所有节点路由表中该节点 N 出现的次数(所有节点包括那些没有响应我们查询的活跃节点).实际网络测量中,搜索器无法获得真实网络在某一时间段内全部节点信息,因此节点的 RSI 指数是无法得到的.RSSI 指数计算那些响应我们查询的活跃节点所返回的应答消息中节点 N 出现的次数,是可测量的,这里给出 RSSI 的形式化定义如下:

$$RSSI(N_i)=\sum_{N_j\in NS_{alive}}InRouteOf(N_i,N_j).$$

并且有:

$$RSI(N_i)>=RSSI(N_i).$$

一般来说,节点有较大的 RSSI 指数意味着该节点广泛地参与了与其他节点的信息交互,因此假设这样的节点在网络中有较长的存活时间.因此,在每次搜索结束时,通过简单地计算可以获得节点的 RSSI 指数,开始一次新的搜索时,每个节点的 RSSI 指数取值为历史记录中最近 n 次搜索该指数的加权均值(由于正常节点的 RSSI 指数变化缓慢,取均值可以一定程度上遏制恶意节点的 sybil 攻击[16]).如果上述均值大于某个节点筛选阈值 L,则该节点视作关于 RSSI 指数是活跃的.这里,阈值 L 的具体取值通过实验获取,使其对活跃节点与非活跃节点区分度最大.下面给出修改后的 Λ 函数的定义以及相关概念的形式化描述:

定义相对于当前搜索的过去连续 n 次搜索的集合为 CS_n,其中,TP_n 为相对于当前搜索时刻的过去 n 次搜索的开始时刻的集合(这里,时刻 t_i 用来惟一标识某一次搜索):

$$CS_n=\bigcup_{t_i\in TP_n}Crawl_{t_i}(\Lambda,\Phi),TP_n=\{t_1,t_2,...,t_n\}.$$

定义布尔函数 $RSSI\_CHECK$,表示某个节点关于 RSSI 指数是否活跃:

$$RSSI\_CHECK(N\in SP_{all\_send},CS_n)=\begin{cases}1, & \sum_{t_i\in TP_n}w_i\cdot RSSI_{t_i}(N)\geqslant L\\ 0, & \sum_{t_i\in TP_n}w_i\cdot RSSI_{t_i}(N)<L\end{cases}.$$

依赖过去 n 次搜索中节点 RSSI 指数的 Λ 函数定义为如下形式:

$$\Lambda(N\in SP_{all\_send},CS_n)=RSSI\_CHECK(N,CS_n).$$

为了查询新加入网络中的 RSSI 较小的节点,仅仅检测 RSSI 指数是不够的,需要同时检查过去 m 次(与上述 RSSI 指数检查中的 n 不同,可以取相同数值)搜索中每个节点的存活情况.由于无法获取节点的真实存活状态,所以用节点针对搜索的应答查询情况来近似模拟.在检查过去 m 次搜索中节点存活状况时,为了避免一些节点因为负载较大,在某几次搜索中没有回复查询而被视作非活跃节点,本文借鉴 Steiner[1]的方法忽略掉连续搜索结果中有限次数的存活性中断,即节点是活跃状态的任意两次搜索中间出现至多 e 次该节点为非活跃状态的搜索.当 e 取值 2 时,我们定义 m 次搜索中某个节点的存活布尔函数 M_CRAWL_ALIVE 为

$$M\_CRAWL\_ALIVE(N,CS)=\forall t_i\in TP(C\_ALIVE_{t_i}(N)=0\rightarrow((i=1\vee i=n)\wedge(C\_ALIVE_{t_{i+1}}(N)=1\vee C\_ALIVE_{t_{i-1}}(N)=1))).$$

经过调整后的 Λ 函数重写如下:

$$\Lambda(N\in SP_{all\_send},CS_m)=RSSI\_CHECK(N,CS_m)\vee M\_CRAWL\_ALIVE(N,CS_m).$$

但是,较长的存活时间并不意味着该节点会以较大的概率回应搜索器的查询请求,因为一部分存活较长时间的节点在搜索时可能已经离开网络,因此,Λ函数存在固有的系统误差,并且这个误差会带来额外的搜索代价.从第 3.5 节针对具体的 KAD 网络实验,可以看到这个误差对搜索结果和性能的影响.

以上的Λ函数选择的是那些在最近几次搜索中积极回应查询请求或者在对等网络中有较长生存期的节点,使用该Λ函数来优化搜索过程存在如下两个制约条件:

1) 开始搜索前,需要使用指定次数的朴素搜索方法以完成 RSSI 和节点活跃率等历史数据指标的计算;

2) 新的搜索只能选择出现过的节点作为查询目标.

由条件 2)可见,上面给出的Λ函数定义是不完备的.由于实际测量中新发现的节点数量所占的比例较小,因此本文拟采取直接查询新发现的节点而不经过Λ函数筛选.Λ函数修改如下:

$$\Lambda(N\in SP_{all\_send},CS_m)=\begin{cases}(RSSI\_CHECK(N,CS_m)\vee M\_CRAWL\_ALIVE(N,CS_m), & \exists N\exists t(t\in TP_n\wedge N\in NS_{all,t})\\ 1, & \text{else}\end{cases}.$$

这里涉及一个历史数据窗口长度 n 以及更新频率的选取问题,优化所需的历史数据要覆盖一定长度,从而弥补单次搜索由于自身性能限制导致的固有误差以及网络扰动带来的随机误差,保证历史数据能够覆盖这个时间点上大多数的节点.更新频率过快,则带来较大的计算资源消耗;过慢则会导致更新窗口后期的搜索结果性能快速下降.因此,这两个参数都需要针对具体的结构化对等网进行针对性的设置和调优.

2.3.4 搜索过程中节点查询消息Φ函数的优化

Steiner 等人[1]的测量方法中,查询消息函数$\Phi(N)$是固定的 16 个查询消息,这些查询能够获得被查询节点路由表中的大多数节点信息.但是实际测量中发现,最新版本的 KAD 网络对接受查询的速率进行了限制,因此这个方法已经不再适用.周模等人[12]从 KAD 网络的路由存储结构出发,通过计算获得能够覆盖大多数 K 桶的查询消息.但是这些方法过于谨慎地确保了完备性,使得节点查询应答中的节点消息存在大量冗余.

搜索冗余指数 SRI 是关于全部节点的统计数据,难以作为设计$\Phi(N)$的依据.为此,本文引入一个新的测量指标——查询后继活跃率 QDAR(query descendant alive rate),表示在一个节点的一次查询返回的所有节点中活跃节点所占的比率.

定义节点 N 针对一次查询消息 $q\in\Phi(N)$返回的节点集合为 $RS(N,q)$,那么 QDAR 可以表示如下:

$$QDAR(N,q\in\Phi(N))=\{N|N\in RS(N,q)\wedge C\_ALIVE(N)\}.$$

为了测量一个节点的查询回应结果对一次完整搜索构成的影响,这里引入搜索中节点回复一个查询消息的收益的概念 $Q\_Benefit$,定义如下:

$$Q\_Benefit(N,q)=\{N_i\mid\exists j_1,j_2,...,j_t((N_i=N_{j_t})\wedge(N_{j_1}=N)\wedge(\forall m\in\{1,2,...,t-1\}(N_{j_{m+1}}\in R(N_{j_m}))))\},$$

其中,$R(N)$前面已经定义过,用 $RS(N,q)$也可以表示如下:

$$R(N)=\bigcup_{q\in\Phi(N)}RS(N,q).$$

另外,以Φ为节点消息函数时,节点在一次搜索中的全收益 $Q\_Benefit\_All$ 定义如下:

$$Q\_Benefit\_All(N,\Phi)=\bigcup_{q\in\Phi(N)}Q\_Benefit(N,q).$$

基于测量经验和一些基本事实,利用历史搜索结果中节点的 QDAR 指数,可以预测当前和未来搜索中节点回复查询的概率.本文假设节点的 QDAR 指数越高,节点回复查询的收益越大.该假设可分解为如下 4 个条件:

(1) 在时刻 t,若$\Phi_1(N)$包含于$\Phi_2(N)$,则$|Q_Benefit_All(N,\Phi_1)|<=|Q_Benefit_All(N,\Phi_2)|$;

(2) 对于消息 q,如果 $QDAR(N_i,q)>QDAR(N_j,q)$,则$|Q_Benefit(N_i,q)|>|Q_Benefit(N_j,q)|$成立;

(3) $QDAR_{t_1}(N,q)\approx QDAR_{t_2}(N,q)$ (t_1,t_2 分别是同一环境下连续两次搜索的开始时间);

(4) 增加部分选取节点的$|\Phi(N)|$到一定程度后,$Q\_Benefit\_All$ 继续增加而$|NS_{all}|$不再增加.

第 1 个条件是指对于同一节点,搜索全收益是关于$|\Phi(N)|$单调递增的,由 $Q\_Benefit\_ALL$ 定义可以看出该条件是成立的.第 2 个条件是指对于不同的节点和同一个查询消息,搜索收益 $Q\_Benefit$ 关于节点的 QDAR 指数是单调递增的.由于节点的 $Q\_Benefit$ 指数计算较复杂,因此本文采用贪心策略,假定针对全部搜索结果的搜索收

益指标 *Q_Benefit* 满足该条件,该项假设条件只有对于大多数节点成立时才会有效地提高搜索性能,需要通过测量实验予以确认.第 3 个条件是指节点的 QDAR 指数关于搜索序列的变化是缓慢平稳的,这是本文使用历史数据进行预测的前提.最后一个条件是指搜索结果中节点数量比搜索全收益更快地收敛到一个稳定数值.后面两个条件同样依赖于对等网的协议细节以及测量结果来证实.本文后续实验结果证实:该项假设对于大多数节点成立,研究该项假设是否适用于一般结构化对等网络是我们下一步工作内容之一.

2.3.5 优化搜索算法描述

综合前面 4 节的讨论分析,本文提出的结构化对等网的全网搜索方法如下:

改进的搜索算法:

输入:初始启动节点集合 *BootstrapSet*;

RSSI 历史搜索记录数量:n,节点筛选阈值 L,各次搜索中 *RSSI* 指数的加权值 $w_1,w_2,\ldots,w_n$;

节点历史活跃状态记录数量:m;

连续搜索中间的等待时间间隔:t;

输出:结构化对等网中全部节点集的一个搜索抽样.

发送线程算法:

Begin

1. 启动连续 max(m,n)次前述朴素搜索查询算法
2. 计算最近 m 次搜索查询中各个节点的 *RSSI_CHECK* 函数值以及最近 n 次搜索查询中各个节点的 *M_CRAWL_ACTIVE* 函数值,存储到搜索历史优化数据库中.
3. 启动下面循环 *Crawl_ALL* 进行连续多次搜索:

   Crawl_ALL

   Begin

   a) 从前一次搜索历史中选择启动节点;

   b) 使用第 2.3.3 节的Λ函数作为节点选择函数,选择采取基于节点 QDAR 参数的局部贪心策略$\Phi(N)$作为节点消息函数进行搜索;

   c) 更新搜索优化数据库;

   d) 等待指定的时间;

   End

End

3 KAD 网络搜索实验

本文以 KAD 网络为测量对象,实现了前述的搜索算法并开发了搜索工具 KadCrawler,用 IP 地址、KAD 网络 ID 以及 UDP 端口来唯一标识一个节点.搜索结束后,计算出每一个节点的 RSSI 指数、QDAR 指数等相关参数.测量实验均在两台普通 VPS 主机上进行,配置为 512M 内存,100M 共享带宽.

3.1 KAD网络搜索的基本方式

eMule 软件中,KAD 网络协议的实现提供了两种从其他节点获取更多节点信息的方式,即 bootstrap 查询[1]以及 route 查询[14],而已有工作[1,14]并没有对这两种查询方式进行区分.实验数据显示,两种方法的结果在各项统计指标上有着显著的差异.表 2 所示在主机配置和网络状况等条件相似的环境下分别进行这两种查询方式的搜索,route 查询方式消耗的带宽是 bootstrap 方式的 8.6 倍,多消耗 44s 的条件下,多获取了接近 40 万节点,在活跃节点数量上却少了 12 698 个节点.目前,KAD 实现中的 route 查询通过一个反映在线时长的状态参数的过滤自动屏蔽了一些新加入网络的节点,使得这两种查询方式的差异更加明显.

Table 2 Comparison of performance between two search methods

表 2 Route 与 Bootstrap 单次搜索性能比较

搜索方法	节点数量	活跃节点数量	消耗带宽(bytes)	开始时间	消耗时间
route	1 230 545	247 848	2460919×51	2012-8-10 5:45:48	4′37″
bootstrap	833 289	260 546	806631×18	2012-8-10 5:45:47	3′53″

为了比较两种搜索方法在较长时间上的搜索性能,我们从 2012-8-10 5:45~2012-8-10 21:30(UTC 时间)对 KAD 网络进行了 177 次 bootstrap 测量和 97 次 route 测量,全部为针对所有节点的完全搜索测量,结果如图 1、图 2 所示.route 搜索方式在总的节点数量上平均超过 bootstrap 方式 43.5%,而在活跃节点数量上则减少了 9.3%.

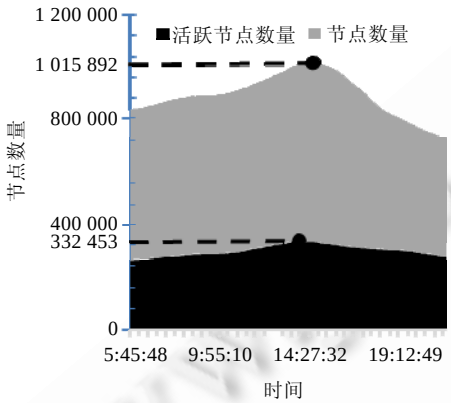


Fig.1 Continuous measurement (bootstrap)

图 1 连续测量(bootstrap 方式)

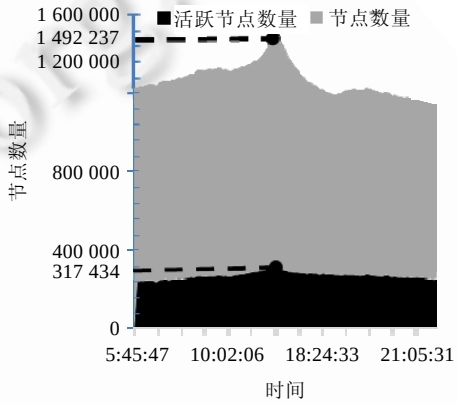


Fig.2 Continuous measurement (route)

图 2 连续测量(route 方式)

由两种搜索方法本身来看,bootstrap 查询结果只能覆盖被查询节点中固定有限的部分,在较短时间(相对于搜索时间)内返回的结果相差很小,因此一般只需要进行一次查询,查询结果很快收敛.Bootstrap 查询获取的节点信息有限,但是却成功搜索到了比 route 查询更多的活跃节点信息;route 查询可以自由选择查询包中的目标节点 ID,因而 Steiner[1]提出的搜索方法依赖于高带宽高速网络环境,通过发出大量覆盖式查询来尽可能地抓取 KAD 网络中的全部节点信息,这种方法在 KAD 网络协议增加查询速率限制以后不再可行.Bootstrap 方式可以获得某一时间段内更多的新节点信息和活跃节点信息,而 route 查询可以获得更多的节点信息.因此,依据搜索目标的不同,我们可以综合两种搜索方式的优点改进搜索过程.本文中则选择使用 bootstrap 方式来验证优化的有效性,后续工作中,我们会针对具体的应用场景综合上述两种方式的优点搭配设置,以达到搜索的最佳性能.

3.2 KAD网络搜索中Λ函数优化

Λ函数利用历史搜索数据中节点的存活状况以及 RSSI 指数作为下一轮搜索中是否查询该节点的依据,因此,有必要对 KAD 网络中节点的存活状况以及 RSSI 指数进行实验测量和分析,以确定优化搜索中各项参数的取值.

我们从 2012-7-1 5:00~2012-7-2 8:00 进行了 400 次朴素搜索,测量搜索中节点存活性中断现象,以及统计搜索中节点存活状态随搜索序列(搜索序列随时间递进)变化的情况.通过对第 1 次搜索得到的节点在后续搜索结果中的状态序列进行统计分析,本文把节点状态按照是否出现在搜索结果中分为两种,即存在和不存在,用 P 和 nP 来表示,其中,节点存在时的状态 P 又可以分为活跃、非活跃以及未匹配(数量较少)这 3 种,分别用 P1,P2,P3 来表示.本文统计了搜索序列中转移到 P1 前保持在 P2 状态至少一定长度时所对应的节点数量,数据显示,节点从 P2 转移到 P1 的情况在节点的状态变化序列中是普遍存在的,有超过 27.27%的节点在搜索结果中出现过这样的状态转移,如图 3 所示,其中,存活性中断长度至多为 2 的节点数量占到了 74%.因此,本文在前面提出的容忍存活中断次数 e 取 2 是在计算代价和效率综合平衡之中比较合理的选择.

本文对搜索序列前 100 个搜索结果的活跃节点数量与搜索序列标号进行了一元线性回归,其中,搜索序列近似代表时间序列,得到如下表达式:$Y=282844-1126X$,Y 代表活跃节点数量,X 代表时间序列.这里,一个单位代表平均的搜索消耗时间(平均为 4′30″).每次搜索消耗都会有 1 126 个节点会由活跃状态转变为非活跃,这样的衰减趋势可如图 4 所示.可见,针对 KAD 网络搜索,有必要在有限次数的优化搜索后对节点优化信息数据库进行更新(参见第 2.3.5 节算法的 3c)部分),添加最近活跃的节点信息,以避免性能的显著下降.

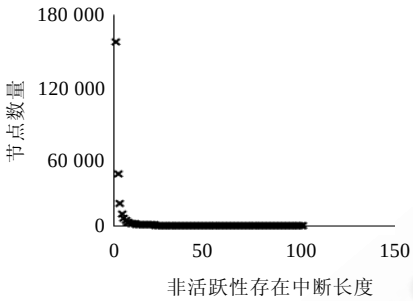


Fig.3 Live interruption analysis in search sequence

图 3 搜索序列中存活性中断现象统计

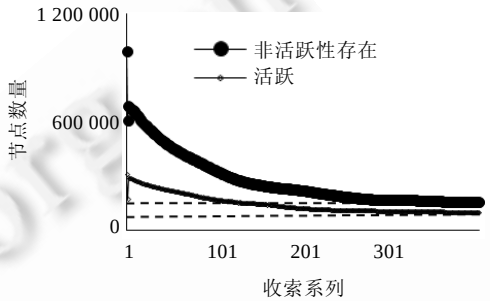


Fig.4 Attenuation effect of node liveness and presence

图 4 节点活跃性与存在性的衰减

从上面的连续搜索结果集中随机抽样一个结果集进行 RSSI 指数的分析,本文统计了 RSSI 指数所对应的节点数量,如图 5、图 6 所示.数据显示,53.1%的活跃节点其 RSSI 指数在 10 以下,而 95.09%的非活跃节点其 RSSI 指数在 10 以下.因此,本文将搜索算法中的 RSSI 阈值设置为 10,这样可以节省与非活跃节点之间浪费的大量通信带宽.

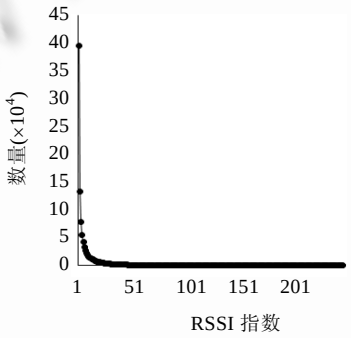


Fig.5 Measurement of node's RSSI

图 5 节点 RSSI 指数测量

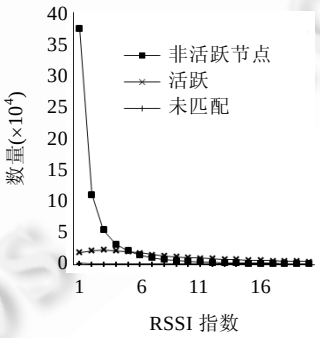


Fig.6 Measurement of nodes's RSSI (0~20)

图 6 节点 RSSI 指数(0~20)测量

从实验结果可以看出 RSSI 指数优化与节点历史活跃性优化之间的关系:如果只有 RSSI 指数的优化过滤,则 53.1%的节点会被过滤掉(RSSI 屏蔽阈值同上,取值为 10);如果只有节点历史活跃性优化,则部分因为负载较高无法及时回复的节点即使有较高的 RSSI 值也会被过滤掉.因此,这两个优化方法是互相补充的,有助于改进搜索过程.

3.3 KAD网络搜索中Φ函数优化

Φ函数利用每个查询节点过去返回结果中活跃节点比率来确定搜索该节点时发送查询请求的数量,即依赖于历史搜索结果中节点的 QDAR 指数.如果向同一个节点发送多个消息,则 QDAR 指数取值为该节点在各次搜索中对应 QDAR 指数的均值.依据优化策略,优先向 QDAR 指数较大的节点发送较多的查询.

从随机抽取的搜索结果样本中,本文计算了 QDAR 指数的节点数量分布.数据显示:单次查询中节点的

QDAR 指数集中在 6 附近,且呈钟形分布,QDAR 指数在 2~11 的节点数量占全部活跃节点数量的 95.8%(搜索采用 bootstrap 方式,协议限制其 QDAR 指数最大为 20),如图 7 所示.约 96%的活跃节点,其 QDAR 指数在 2~11 之间.因此,本文设置Φ函数中 QDAR 指数阈值为 6,即 QDAR 指数不小于 6 的节点会得到更多次数的查询.活跃节点的 QDAR 指数与 RSSI 指数的斯皮尔曼相关系数[17]为 0.19,说明两个指数之间存在一定的正相关性,其关系如图 8.下一步工作中,我们将考虑利用该相关性来精简优化过程.

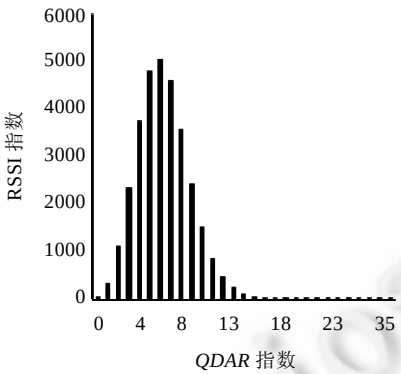


Fig.7 Node count with respect to QDAR

图 7 QDAR 指数对应的节点数量分布

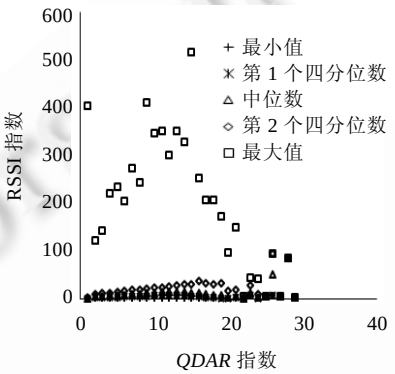


Fig.8 RSSI value with respect to QDAR

图 8 QDAR 指数对应的 RSSI 指数分布

对于 KAD 网络两种不同的搜索方式,*Φ*函数的具体查询方式也不同:对于 route 查询,QDAR 指数较大,则发送较多数量的查询并且覆盖不同的 K 桶[7];对于 bootstrap 查询,则采取在搜索结束阶段对于 QDAR 指数较大的节点进行一次重复查询,以获得额外的节点信息.

3.4 KAD网络搜索优化方法小结

本文提出的方法需要依据结构化对等网的具体结构设计进行 3 个方面的设置,即选择具体搜索方式(bootstrap 或 route)、*Λ*函数、*Φ*函数.由于 KAD 网络中 route 查询代价高,性能指标方差较大,在不同机器和带宽条件下搜索性能差距难以有效控制和进行比较,因此本文选择 bootstrap 方式来验证比较优化算法的性能改进.对*Λ*函数来说,历史信息(包括 RSSI 和节点活跃性)窗口都选为 3,节点 RSSI 指数为各次取值的均值加权和并且屏蔽阈值设置为 10.在实验评估中,为了便于观察优化性能随时间变化的情况,不启动优化信息定时更新机制(见第 2.3.5 节算法的 3c)部分).*Φ*函数则同第 3.3 节,QDAR 指数大于等于 6 的节点获得一次额外的查询.

3.5 KAD网络搜索优化效果评测

对于 KAD 网络来说,搜索性能的评测采用 4 项评价指标,包括搜索消耗的时间、带宽、搜索结果中无重复的节点数量以及节点活跃率.这里没有使用搜索冗余指数 SRI 这一指标,因为 KAD 网络中每一次查询返回的其他节点数量是固定不变的,因此可以通过消耗带宽和获得无重复节点数量这两个指标来计算得到.对于其他返回非固定数量节点信息的网络来说,SRI 指数是衡量搜索优化方法的重要指标.

Table 3 Performance comparison of naive search and optimized search

表 3 单次朴素搜索与优化搜索性能比较

搜索方法	节点数量	活跃节点数量(活跃率)	消耗带宽(bytes)	开始时间(北京时间)	消耗时间
朴素	894 315	271 822 (30.3%)	867438×18	2012-8-10 15:48:05	3′16″
优化	897 908	275 850 (30.7%)	495258×18	2012-8-10 15:47:25	3′34″

本文搜索算法采用的各项参数依照第 3.3 节的分析结果来设置,朴素搜索与优化搜索在大致相同的时间开始进行一次搜索,其实验结果见表 3.我们看到:优化方法在结果的节点数量、节点活跃率上大致相当,消耗带宽

减少 43%,但是时间上多消耗了 18s.

在初次比较的基础上,我们随后进行了 10 小时的连续搜索测试,其中 160 次朴素搜索,99 次优化搜索.搜索完毕后进行了性能相关的各项指标的比较,以观察没有优化数据更新的搜索算法性能的变化情况,如图 9~图 12 所示(为了显示不同方法搜集活跃节点绝对数量的能力,图 10 中用活跃数量代替活跃率).

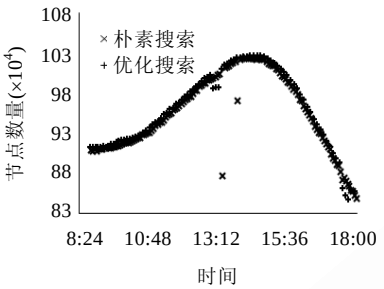


Fig.9 Comparison of total node size

图 9 搜索结果节点数量对比

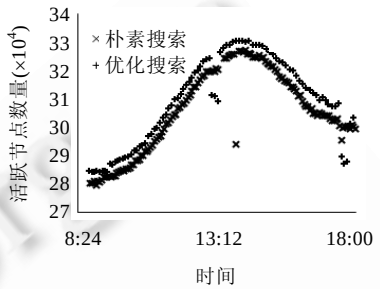


Fig.10 Comparison of active node size

图 10 搜索结果活跃节点数量对比

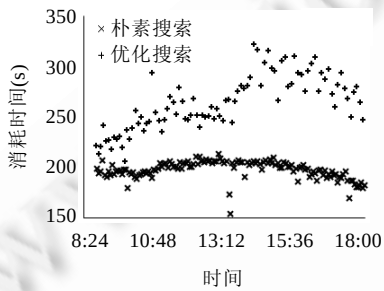


Fig.11 Comparison of time consumed

图 11 搜索结果消耗时间对比

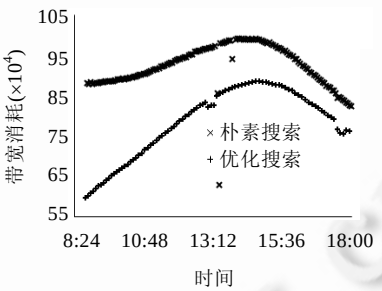


Fig.12 Comparison of bandwidth consumption

图 12 搜索结果带宽消耗对比

结果显示:在连续运行的较长一段测试时间内,优化搜索获得的节点信息和活跃节点数量与朴素搜索大致相同.图 10 中,优化方法在活跃节点数量上略微提高,优化效果并不显著,可能是搜索环境引入的固有系统误差.

实验结果显示,优化方法消耗了较多的时间,如图 11 所示.为了验证该差异是否为不同环境造成的固有系统误差,本文在两个不同的环境下分别进行了朴素搜索测试.令上述实验中朴素搜索运行环境为 *A*,ip 地址为 209.140.*.*;优化方法运行环境为 *B*,ip 地址为 66.175.*.*.我们测试了 Amazon EC2 分布在爱尔兰、巴西、日本和美国的主机分别与 *A*,*B* 主机之间的链路性能(网络链路性能检测工具 iperf),没有发现显著的差异,可有效利用的带宽均为 74.65Mbit/s.朴素搜索测试的实验结果如图 13 所示,可见 *B* 条件下搜索时间显著高于 *A*.经过计算, *B* 条件下搜索时间的标准差为 13.861,而 *A* 条件下标准差为 37.591.可见,使用相同的搜索方法在 *A* 环境下搜索时间更少,稳定性更好.造成这种现象可能的原因是主机服务商超售主机和带宽的差异造成的.由图 11 和图 13 的数据对比可以看出,优化方法在初期的搜索过程中有效地提高了搜索速度,减少了搜索时间,部分弥补环境 *B* 性能的不足.在实际的搜索测量中,建议使用一个固定长度更新窗口,在不造成过大计算负载的同时不断更新节点优化数据库,确保搜索器保持良好的性能.若将更新窗口设置为 4,实验数据显示,优化方法将消耗的带宽平均降低了 41%,搜索时间平均减少了 9%,而搜索结果中节点数量平均下降了不到 0.2%.

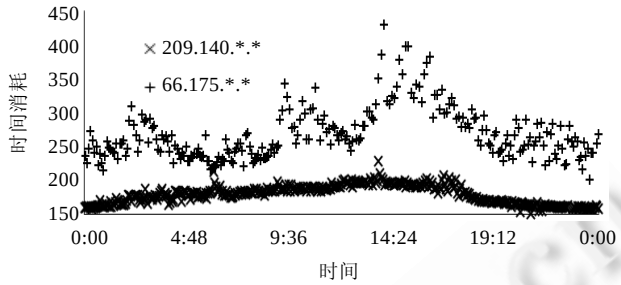


Fig.13 Comparison of search time with naive search method in two different environment

图 13 朴素搜索在不同环境下的消耗时间对比

3.6 KAD网络测量结果分析

我们从 2012-8-9 00:00~2012-8-12 00:00 分(北京时间)对 KAD 网络进行了 1 564 次 bootstrap 方式地连续测量,全部为针对所有节点的完全搜索测量.

节点数量和分布的测量结果如图 14 所示,数据显示,节点数量的最高值在 1 048 459(KAD 网络中 bootstrap 查询在较短时间段内仅能返回固定的 20 个节点信息).节点数量按照时间的变化趋势以一天为单位呈现出一定的周期性,达到峰值的时间集中在每天的 21:50~22:10 之间(北京时间).活跃节点占全部节点的比率平均为 33.27%.而按照国家和区域来划分,样本的统计分析显示:IP 为中国区域(包括中国大陆、港澳台等地区)的活跃节点比率平均为 16.99%,而非中国区域相应比率为 61.68%,如图 15 所示.造成区域间活跃节点比率显著差异的可能原因是中国区域的某些 KAD 客户端(如迅雷、QQ 旋风等)修改了协议实现,拒绝了搜索器的查询请求.如何兼容可能的协议变异,或者依据国家地区类别分别来完成搜索,是我们下一步工作内容之一.

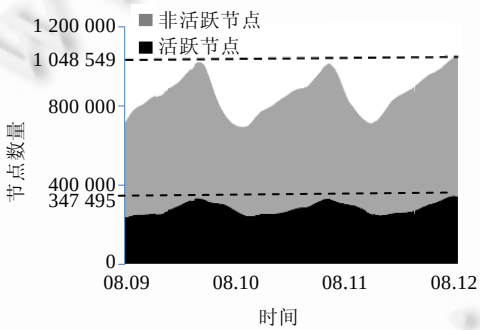


Fig.14 Continuous search result of KAD network

图 14 KAD 网络连续搜索

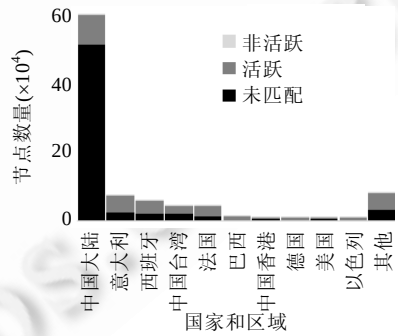


Fig.15 Node count in different areas

图 15 各个区域节点数量统计

最后,通过对一个随机抽取的测量样本数据(总共包含 994 532 个节点,其中活跃节点为 312 909 个)进行统计分析,发现节点重名现象在 KAD 网络中普遍存在,有重复 ID 的节点数量占总数的 33.4%.但是与 JieYu[14]研究工作得到的数据相比,我们发现有两点主要变化:首先,发生 ID 重复的节点中非活跃节点比率显著增加,占到了 99%,其中,IP 为中国大陆的节点占到了 89.8%;其次,单个重复 ID 拥有节点的数量大幅下降,最大值为 1 746.造成这种现象的原因,我们认为:一方面是符合约定规范的客户端数量减少,大量的不规范客户端导致搜索结果中非活跃节点比例增加;另一方面是,部分节点利用 KAD 协议的缺陷伪造了大量重复 ID.

4 总结与展望

本文对结构化对等网的搜索过程进行了形式化建模和分析,提出了节点路由延展指数以及查询有效比率的概念,分析了这些指标与搜索结果之间的关系,并以此为基础设计了一套结构化对等网节点信息搜索优化方

法.我们对典型的结构化对等网 KAD 网络进行大量测量,分析总结了优化算法相关参数和指标的特征规律,并且通过在搜索过程中应用优化方法进行了实验验证.结果表明,优化方法可以减少带宽占用和搜索时间,并且保持搜索结果基本完整.

本文对 KAD 网络进行了初步测量和分析,发现了一些新的现象.后续工作中,我们将继续针对 KAD 网络 ID 重名现象进行分析,发掘其背后隐藏的文件共享相关的语义信息,并拓展相关研究方法到 BT DHT 等流行的结构化对等网络中.同时,一些移动互联网时代新的应用也同样值得关注,如基于最新 Web 技术的 WebRTC 项目拓展了结构化对等网的应用领域,将来很有希望成为新的研究热点.此外我们注意到,结构化对等网在安全领域的应用有待深入研究.Storm,TDL4 等先后出现的借助结构化对等网进行去中心化控制的僵尸网络程序,其隐蔽性和匿名性都很强,对网络安全提出了很大的挑战,下一步我们也将对此进行研究和探索.

致谢 在此,我们向对本文的工作给予支持和建议的同行,尤其是中国科学院软件研究所的苏璞睿研究员、和亮博士以及研究小组的各位老师和同学表示感谢.

References:

[1] Steiner M, En-Najjary T, Biersack EW. Long term study of peer behavior in the KAD DHT. IEEE/ACM Trans. on Network, 2009, 17(5):1371−1384. [doi: 10.1109/TNET.2008.2009053]

[2] Liu XT, Meng T, Cai K, Cheng XQ. Rainbow: A robust and versatile measurement tool for Kademlia-based DHT networks. In: Proc. of the Int'l Conf. on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2010). 2010. [doi: 10.1109/PDCAT.2010.18]

[3] Daniel S, Reza R. Understanding churn in peer-to-peer networks. In: Proc. of the 6th ACM SIGCOMM Conf. on Internet Measurement. Rio de Janeriro: ACM Press, 2006. 189−202. [doi: 10.1145/1177080.1177105]

[4] Sen S, Wang J. Analyzing peer-to-peer traffic across large networks.Networking. IEEE/ACM Trans. on Networking, 2004,12(2): 219−232. [doi: 10.1109/TNET.2004.826277]

[5] Gummadi KP, Dunn RJ, Saroiu S, Gribble SD, Levy HM, Zahorjan J. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In: Proc. of the 19th ACM Symp. on Operating Systems Principles. Bolton Landing: ACM Press, 2003. 314−329. [doi: 10.1145/945445.945475]

[6] Wang Y, Yun XC, Li YF. Measuring and characterizing topologies of P2P networks. Ruan Jian Xue Bao/Journal of Software, 2008,19(4):981−992 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/981.htm [doi: 10.3724/SP.J.1001.2008.00981]

[7] Maymounkov P, Mazières D. Kademlia: A peer-to-peer information system based on the XOR metric. Revised Papers from the 1st Int'l Workshop on Peer-to-Peer Systems. Springer-Verlag, 2002. 53−65. [doi: 10.1007/3-540-45748-8_5]

[8] Stoica I, Morris R, Karger D, Kaashoek MF, Balakrishnan H. Chord: A scalable peer-to-peer lookup service for Internet applications. In: Proc. of the 2001 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications. San Diego: ACM Press, 2001. 149−160. [doi: 10.1145/383059.383071]

[9] Rowstron A, Druschel P. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: Proc. of the IFIP/ACM Int'l Conf. on Distributed Systems Platforms. Heidelberg: Springer-Verlag, 2001. 329−350.

[10] Ratnasamy S, Francis P, Handley M, Karp R, shenker S. A scalable content-addressable network. In: Proc. of the 2001 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications. San Diego: ACM Press, 2001. 161−172. [doi: 10.1145/964723.383072]

[11] Liu XT, Gong CC, Liu Y, Bai S. Peer resource management and analysis in KAD network. Journal of Chinese Information Processing, 2010,24(6):85−91 (in Chinese with English abstract).

[12] 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.

[13] Wu Q, Chen XS. Advanced distributed crawling system for KAD network. Computational Information Systems, 2011,7(3):677−684.

[14] Yu J, Fang CF, Xu J, Chang EC, Li ZJ. ID repetition in KAD. In: Proc. of the 9th IEEE Int'l Conf. on Peer-to-Peer Computing (P2P 2009). 2009. [doi: 10.1109/P2P.2009.5284551]

[15] Yu J, Li ZJ. Active measurement of routing table in KAD. In: Proc. of the 6th IEEE Conf. on Consumer Communications and Networking Conf. Las Vegas: IEEE Press, 2009. 1252−1256.

[16] Douceur JR. The sybil attack. In: Proc. of the 1st Int'l Workshop on Peer-to-Peer Systems. Springer-Verlag, 2002. 251−260.

[17] Spearman C. The proof and measurement of association between two things. The American Journal of Psychology, 1987,100(3/4): 441−471.

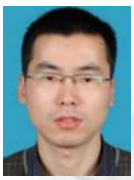
附中文参考文献:

[6] 王勇,云晓春,李奕飞.对等网络拓扑测量与特征分析.软件学报,2008,19(4):981−992. http://www.jos.org.cn/1000-9825/19/981.htm [doi: 10.3724/SP.J.1001.2008.00981]

[11] 刘祥涛,龚才春,刘悦,白硕.Kad 网络节点资源探测分析.中文信息学报,2010,24(06):85−91.

闫佳(1986－),男,山西长治人,博士生,主要研究领域为网络测量和分析,僵尸网络.
E-mail: yanj@is.iscas.ac.cn

苏璞睿(1976－),男,博士,研究员,主要研究领域为恶意代码分析,入侵检测.
E-mail: supurui@is.iscas.ac.cn



应凌云(1982－),博士,助理研究员,主要研究领域为网络安全,恶意代码分析,P2P 网络.
E-mail: yly@is.iscas.ac.cn

冯登国(1965－),男,博士,研究员,博士生导师,CCF 高级会员,主要研究领域为密码学,网络与系统安全.
E-mail: feng@is.iscas.ac.cn

刘海峰(1976－),博士,研究员,主要研究领域为信息安全测评,软件测试.
E-mail: bjcert@bjeit.gov.cn