

## 面向非结构化 P2P 网络的双向随机漫步搜索机制\*

马文明<sup>1,2+</sup>, 孟祥武<sup>1,2</sup>, 张玉洁<sup>1,2</sup>

<sup>1</sup>(智能通信软件与多媒体北京市重点实验室(北京邮电大学),北京 100876)

<sup>2</sup>(北京邮电大学 计算机学院,北京 100876)

### Bidirectional Random Walk Search Mechanism for Unstructured P2P Network

MA Wen-Ming<sup>1,2+</sup>, MENG Xiang-Wu<sup>1,2</sup>, ZHANG Yu-Jie<sup>1,2</sup>

<sup>1</sup>(Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia (Beijing University of Posts and Telecommunications), Beijing 100876, China)

<sup>2</sup>(School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China)

+ Corresponding author: E-mail: mwmytu@gmail.com, http://www.bupt.edu.cn

Ma WM, Meng XW, Zhang YJ. Bidirectional random walk search mechanism for unstructured P2P network. *Journal of Software*, 2012, 23(4): 894-911. <http://www.jos.org.cn/1000-9825/4086.htm>

**Abstract:** The improvements of random walk search mainly depend on allocating weight for neighbor peers, which is always incurred on a high overhead and are not very helpful for rare items. This paper proposes a bidirectional random walk search mechanism (short for BRWS) for unstructured P2P network, according to the analysis of basic properties about random walk as well as the special property that random walk tends toward high degree nodes. The mechanism is proved theoretically in this paper, and can improve search success rate, including searching for rare items. It also has a high tolerance for churn. In the static and dynamic environment, comparisons were made among Random Walk, APS (adaptive probabilistic search), PQR (path-traceable query routing), P2PBSN (peer-to-peer based on social network) and BRWS based on three topologies: Random graph, scale free, small world. The experimental results show that BRWS can actually improve the search success rate with lower overhead even when searching rare resources. The method proposed in this paper can apply in P2P file sharing networks.

**Key words:** peer-to-peer; search; random walk; tolerance for churn; topology

**摘要:** 为解决目前 Random Walk 改进算法中过于依赖历史搜索记录而导致动态网络环境下搜索命中率低、网络开销过高和稀有资源的搜索成功率提高不明显等问题,通过分析随机漫步的基本性质和易转向高度数节点的搜索特性,提出了一种双向随机漫步搜索机制——BRWS(bidirectional random walk search),并证明了其能够提高包括稀有资源在内的搜索成功率,抗扰动性强.分别在静态和动态网络环境中,将 Random Walk, APS(adaptive probabilistic search), PQR(path-traceable query routing), P2PBSN(peer-to-peer based on social network)和 BRWS 基于 Random Graph、Scale Free 网络、Small World 网络 3 种拓扑进行了对比实验.结果表明, BRWS 可以以较少的网络搜索代价,极大地提高搜索成功率;并在动态网络环境中,对稀有资源的搜索成功率也有显著提高.所提出的方法可适用于 P2P 文件分发网络应用中.

\* 基金项目: 国家自然科学基金(60872051); 中央高校基础研究基金(2009RC0203); 北京市教育委员会共建项目  
收稿时间: 2010-12-21; 修改时间: 2011-04-28; 定稿时间: 2011-06-20

关键词: Peer-to-Peer;搜索;随机漫步;抗扰动;拓扑

中图法分类号: TP393 文献标识码: A

过去 10 年,P2P(peer-to-peer)网络在许多领域中都得到了广泛应用,P2P 网络总体上分为两种类型:结构化网络(structured network)和非结构化网络(unstructured network).结构化类型的网络主要有 Tapestry<sup>[1]</sup>,Chord<sup>[2]</sup>,PAST<sup>[3,4]</sup>,Viceroy<sup>[5]</sup>等.尽管结构化网络可以有效地定位资源,但并不支持关键字搜索,并且节点退出和离开的信息开销大,扰动容忍度差,在实际网络中很少采用这种结构.非结构化网络抗扰动能力强,能够容忍大量节点的加入和离开,支持关键字搜索,当前普遍采用的都是该结构.

非结构化网络(decentralized and unstructured network)的典型代表是 Gnutella,其采用的 Flooding 方法搜索代价高,不可扩展;随着网络规模的增大,性能急剧下降.Random Walk 是代替 Flooding 的一种重要算法,在高度连接性 P2P 网络中性能较为优越.由于邻居的选择依然带有随机性,Random Walk 的改进算法虽然能够降低这种随机性,为每个邻居节点分配一定的权值,选择可能使搜索成功率较高的节点进行转发.但目前,大多数方法抗扰动能力差,并且不支持稀有资源的搜索.

本文提出了一种双向 Random Walk 的搜索机制 BRWS(bidirectional random walk search),利用 Random Walk 易转向高度数节点的特性,将逆向搜索嵌入到正向搜索中形成双向搜索,在提高命中率的同时,又不会产生过多的网络负担.在双向搜索中,检查中途经过的节点资源的变化,将变化信息加入到转发消息中加以传递,进而做出更新.由于大部分随机漫步转发到高度数节点的概率较高,从而能够获得及时的更新,避免搜索的失败,具有较好的抗扰动性.本文的主要贡献是:(1) 分析了正向随机漫步的基本性质、易转向高度数节点的特性,从理论上予以证明;(2) 分析了单纯逆向搜索的缺点,并根据随机漫步的上述特性提出了双向搜索研究思路,并在理论上证明了可以极大提高包括稀有资源在内的搜索成功率,抗扰动能力强;(3) 给出了 BRWS 搜索机制的数据结构和搜索流程,并通过实验证明了 BRWS 在总体搜索、抗扰动性和稀有资源搜索方面性能的优越性,分析了存储开销和路由转发表的可靠性.

本文第 1 节对 Random Walk 算法及其改进的相关研究进行阐述.第 2 节分析随机漫步的基本性质、易转向高度数节点的特点,并给出 BRWS 搜索机制的设计思路和理论分析,介绍 BRWS 的数据结构和搜索流程.第 3 节在静态和动态网络环境下,基于 Random Graph、Scale Free 网络和 Small World 网络这 3 种拓扑,与传统方法进行各种比较.第 4 节总结本文工作,并对未来的研究进行展望.

## 1 相关工作

针对原始 Flooding 算法的缺点,一些研究对其进行了改进.文献[6]采用了迭代加深方法,每次通过增加 TTL 长度,直到搜索成功.文献[7]提出了与其类似的膨胀环的搜索算法,证明了与固定 TTL 长度的 Flooding 算法相比,能够大为降低网络的负载,但是并不能消除重复消息过多的问题.

文献[8]提出了 Modified BFS 方法,当搜索成功时,回复消息会沿着搜索路径原路返回,中间所经过的节点都记录下(query,peer),query 代表请求的资源,peer 代表提供资源的节点.每个节点根据这些记录建立节点偏好,每种请求都有各自的偏好,并按照偏好程度对节点排序,当有符合的请求时,按照对应的偏好信息,选择前几位的节点进行转发.这种方法只记录成功的搜索,一旦所依赖的资源发生变化,中途经过的节点频繁退出,将无法得到反馈调整记录,节点的偏好准确度就会大为降低.

Random Walk 在统计学、物理学、人工智能、机器视觉、人口变化和流行病学等领域有着广泛的应用,文献[8]将其引入到 P2P 搜索中,证明其能够同时降低搜索网络负担和重复消息.文献[9]首次对 Random Walk 进行了理论分析,证明了网络拓扑的变化对 Random Walk 的性能影响不大.

文献[10]提出了一种 Adaptive Probabilistic Search(APS)方法,利用节点的索引值,求出每个节点的转发概率,索引值的变化有两种策略:optimistic 和 pessimistic.基于 optimistic,沿着搜索路径增加每个节点的索引值,只有当失败时,才按原路返回,重新减少索引值;基于 pessimistic 则正好相反.这种方法很难控制采用哪种策略,容

易引起过多的网络负担.另外,当邻居节点具有很高的索引值时,一旦沿着其相关路径查找的资源发生变化,则必须忍受多次失败反馈以更新索引值.

文献[11]提出了一种 ARW 方法,其主要思想是,实行 Walker Self-Replication,每条 Walker 如果在 TTL 时间内没有找到所需要的资源,则继续执行  $K$  个 Walker,直到在规定时间内停止.这种策略其实是介于 Random Walk 和 Flooding 中间的一种策略,虽然成功率有所提高,但所消耗的网络负担仍然很大.文献[12]提出了一种 Enhanced Random-Walk 方法,该方法将自己的文件列表以 Flooding 方式进行传播来提高搜索成功率,具有不可扩展性.文献[13]提出了一种基于动态 TTL 的搜索算法,该算法主要利用缓存记录,在动态网络环境下,这些记录仍具有不可靠性,其效果并不明显.

文献[14]针对传统随机漫步算法的不足,提出了一种基于 DDBF 技术的概率搜索小组 PST 算法,这种算法实现了多个小组之间的互相协同的并行搜索.其缺点是需要各节点首先发布本节点的资源共享信息,另外,在快速变化的网络环境中,很难捕捉资源信息的分布情况.

文献[15]提出了一种基于社会网络特征的 P2P 内容定位策略,它利用社会网络基本原理,通过社会网络特征,搜索过程中转发至信任度较高的节点.这种方法适合特殊条件的网络拓扑,由于仍然依赖过多的历史记录,在动态变化的网络环境中很难发挥效用.本文将文献[15]中提出的方法简称为 P2PBSN(peer-to-peer based on social network).

文献[16]提出了一种可追溯路径的搜索机制 PQR(path-traceable query routing),每个节点保存一种数据结构 Traceable Gain Matrix(TGM),当搜索成功时,逆向沿着经过的节点,记录路径和索引值;一旦搜索失败,所有包含失败路径的节点,删除与其有关的记录.转发消息时,根据 TGM 数据计算每个邻居的转发概率.该方法可以解决 APS 方法中一旦搜索失败,高索引值节点更新慢的问题.但这种机制的网络开销仍然很大,在  $K$  和 TTL 较小的情况下才比较适用.

文献[17]提出了一种基于指数衰减的 Bloom Filter 的 Scalable Query Routing(SQR)转发策略,节点通过建立和更新 SQR,关于这个节点的资源信息随着距离的增加而呈指数衰减.而这个过程是独立与搜索的,信息的准确性取决于更新的速度,而且更新频繁,这种独立过程造成了过多的网络负载.文献[18]提出了与本文类似的 piggybacks 方式,主要用于动态资源的搜索,并以获取 CPU 资源为例.它仅仅专注于信息的及时性,因 CPU 周期动态变化频繁,需要及时获取这些变化才能做出准确的邻居选择.

上述研究大都围绕如何选择邻居节点进行转发的问题,因而不可避免地需要对邻居节点进行优先排列,其偏好选择依赖于历史搜索记录.P2P 网络本身具有扰动性,一方面,邻居节点不断发生变化,另一方面,资源状态不断发生变化,这些所依赖的历史记录,可靠性不高,总体上,该算法效果比原始的 Random Walk 改善得不明显.本文与上述研究的区别在于,摒弃了这种依赖历史搜索记录对邻居节点进行排列的思想,利用随机漫步的搜索特性,在请求搜索资源的同时,资源逆向搜索请求,并且支持稀有资源的搜索,抗扰动能力强.

## 2 双向随机漫步

### 2.1 基本定义

**定义 1(P2P 网络).** 定义为一个六元组  $PN=(N,K,T,M,R,S)$ ,其中: $N$  表示 P2P 网络中节点的集合, $K$  表示节点的类型集合, $K=\{k_1,k_2,\dots,k_n\}$ , $N=\{Node|\exists!k_i\in K,Kind(Node)=k_i\}$ ; $T$  表示网络拓扑结构; $M$  表示网络中资源的种类集合, $R$  表示所有资源副本的集合, $M=\{m_1,m_2,\dots,m_n\}$ , $R=\{r|\exists!m_i\in M,Kind(r)=m_i\}$ ; $S$  表示搜索, $S=(Q_i,m_i,F,H_i,O_i)$ , $Q_i\subseteq N$  是发起搜索的节点集合, $m_i$  是与搜索有关的资源, $F$  代表具体搜索算法, $H_i$  为搜索命中结果的集合, $H_i=\{\langle Node_s,N_d \rangle|Node_s\in Q_i,N_d\subseteq N,Node_s\notin N_d\}$ ,且  $F(Node_s,m_i,\dots)=N_d$ , $O_i$  表示搜索的消息转发数,也即  $Q_i$  条搜索所经过的节点的数量.

对于非结构化的纯 P2P 网络, $|K|=1$ ,不同的搜索算法, $F$  所具有的参数也有所不同.网络拓扑目前主要分为 3 类:Random Graph、Scale Free 网络和 Small World 网络.现实网络中,Scale Free 类型居多.

**定义 2(资源占有率).** 对于某种资源  $m_i$ ,设  $R_i=\{r\in R,Kind(r)=m_i\}$ ,这种资源的占有率定义为

$$r_i = \frac{|R_i|}{|N|} \quad (2-1)$$

定义 3(资源请求占有率). 对于资源  $m_i$ , 假设  $Q_i$  为发起请求此类资源的节点集合, 这种资源的请求占有率定义为

$$q_i = \frac{|Q_i|}{|N|} \quad (2-2)$$

定义 4(网络开销). 由两项指标来体现: 所产生的消息数  $O_i$  以及产生这些消息所消耗的网络带宽  $TO_i$ , 并且

$$TO_i = \sum_{k=1}^{O_i} \text{Message}_k \text{ Bytes} \quad (2-3)$$

上述含义是  $O_i$  个消息所占用的字节数.  $O_i$  是 P2P 应用中的一项重要指标, 最显著的开销是网络中计算机的处理负担, 因为需要不断地中断来处理消息. 因此, 本文从这两个角度来衡量整个网络开销.

定义 5(随机漫步).  $S = \langle Q_i, m_i, F, H_i, O_i \rangle$ , 其中,  $F(\text{Node}, m_i, K, TTL) = N_d, K$  表示漫步数,  $TTL$  表示生存时间.

定义 6(正向随机漫步搜索). 随机漫步中  $S = \langle Q_i, m_i, F, H_i, O_i \rangle$ , 若节点集合  $Q_i$  表示发起请求此类资源的集合,  $H_i$  表示拥有资源  $m_i$  的集合, 即采用 Pull 的方式获取资源  $m_i$  的搜索过程, 称为正向随机漫步搜索.

定义 7(逆向随机漫步搜索). 随机漫步中  $S = \langle Q_i, m_i, F, H_i, O_i \rangle$ , 若节点集合  $Q_i$  表示发起推送此类资源的集合,  $H_i$  表示需要此类资源的节点集合, 这种采用 Push 方式的搜索过程, 称为逆向随机漫步搜索.

## 2.2 主要思想

双向随机漫步的主要思想是: (1) 将逆向搜索嵌入到正向搜索中形成双向搜索; (2) 正向搜索负责传递资源, 可以提高副本占有率; (3) 逆向搜索查找请求, 提高命中率; (4) 正向搜索负责传递资源修改信息, 提高抗扰能力; (5) 利用随机漫步易转向高度数节点的特性, 高度数节点能够接受到较多的资源信息和请求, 作为信息交换的枢纽, 且可获得及时的更新; (6) 逆向搜索嵌入到正向搜索中, 在提高命中率的同时, 不会产生不必要的网络负担.

下面首先探讨正向随机漫步和逆向搜索的性质, 得出一些重要结论; 然后给出双向随机漫步的设计方法, 并予以理论分析.

## 2.3 正向随机漫步搜索性质

### 2.3.1 搜索特性

定理 1. 在  $|Q_i|=1$  的搜索过程中, 假定  $|N|$  和  $r_i \neq 0$  保持不变, 则此搜索的成功率  $SR_i$  及消息数量期望值  $E(O_i)$  分别为

$$SR_i = 1 - (1 - r_i)^{K \cdot TTL} \quad (2-4)$$

$$E(O_i) = \frac{K}{r_i} (1 - (1 - r_i)^{TTL}) \quad (2-5)$$

证明: 由文献[9]可知, 随机漫步每经过一个节点都可以认为是从均匀分布的节点样本空间中的独立抽样, 由独立性可知, 随机漫步每一步转发成功率都为  $r_i$ .  $K \cdot TTL$  步转发过程可看作几何分布, 每条搜索最终成功的概率为  $1 - (1 - r_i)^{K \cdot TTL}$ . 考虑每条漫步的消息数  $O_{i,j} (1 \leq j \leq K)$ , 最终搜索失败的概率为  $(1 - r_i)^{TTL}$ , 在  $TTL=0$  时刻成功的概率为  $r_i(1 - r_i)^{TTL-1}$ , 则  $P\{O_{i,j}=TTL\} = (1 - r_i)^{TTL-1}$ . 对于  $O_{i,j}=l < TTL$ , 有  $P\{O_{i,j}=l\} = r_i(1 - r_i)^{l-1}$ , 由此推得  $O_i$  的期望值为

$$E(O_i) = K \cdot E(O_{i,j}) = K \cdot \sum_{l=1}^{TTL} l \cdot P\{O_{i,j}=l\} = K \cdot \left( \sum_{l=1}^{TTL-1} l \cdot r_i(1 - r_i)^{l-1} + TTL \cdot (1 - r_i)^{TTL-1} \right) = \frac{K}{r_i} (1 - (1 - r_i)^{TTL}). \quad \square$$

根据定理 1, 增加  $K$  和  $TTL$  可以提高搜索成功率, 且两者对成功率的影响相同, 但同时增加了产生的消息数量,  $K$  对网络负载的影响要远大于  $TTL$ , 因此随机漫步尽量采用较小的  $K$  值, 下面考虑  $r_i$  的影响.

定理 2. 对定理 1 中的搜索, 保持  $K$  和  $TTL$  不变, 且  $TTL \geq 2$ , 增加  $r_i$  能够同时提高搜索成功率  $SR_i$ , 降低  $O_i$  期望值  $E(O_i)$ .

证明:提高搜索成功率显而易见,下面主要证明如何降低网络负载,设  $f(r_i) = E(O_i) = \frac{K}{r_i}(1 - (1 - r_i)^{TTL})$ , 其一阶导数  $f'(r_i) = \frac{K \cdot ((1 - r_i)^{TTL-1} \cdot (TTL \cdot r_i + 1 - r_i) - 1)}{r_i^2}$ , 设  $z(r_i) = (1 - r_i)^{TTL-1} \cdot (TTL \cdot r_i + 1 - r_i)$ , 在  $r_i \in (0, 1)$  范围内,  $z(r_i)$  的一阶导数  $z'(r_i) = (TTL - 1)(1 - r_i)^{TTL-2} \cdot TTL \cdot r_i < 0$ , 有  $z(r_i) < z(0) = 1, f'(r_i) < 0, E(O_i)$  是关于  $r_i$  的递减函数.  $\square$

由定理 2 可知,  $r_i$  是提高搜索效率的关键因素. 实际网络环境中,  $r_i$  不断变化, 考虑一段时间  $t$ , 发起并完成  $|Q_i| > 1, |Q_i| < |N|$  条搜索, 搜索前资源占有率为  $r_i(0)$ , 可近似认为搜索过程中每条搜索的成功率为  $1 - (1 - r_i(0))^{K \cdot TTL}$ . 那么,  $|Q_i|$  条搜索成功的期望值为  $E(Q_i) = |Q_i|(1 - (1 - r_i(0))^{K \cdot TTL})$ , 网络负载期望值为  $E(O_i) = \frac{|Q_i| \cdot K}{r_i(0)}(1 - (1 - r_i(0))^{K \cdot TTL})$ . 此

时, 资源边缘率变为  $r_i(t)$ . 提高  $r_i$  一方面由  $|Q_i|$  的数量决定, 另一方面可以在搜索基础上增加副本或索引复制. 文献 [7] 提出了一种路径复制方法, 即搜索成功后, 沿搜索路径进行索引复制. 这种方法的缺点是针对稀有资源搜索的帮助不大, 且增加了网络负载. 本文所提双向搜索的目的就是, 在保证网络负载不过多加重的情况下提高搜索成功率. 为此, 需要对随机漫步的转发特性进行探讨.

### 2.3.2 转发特性

给定度数分布  $(k, p_k)$ , 根据文献 [19], P2P 网络  $PN$  的拓扑结构  $T$  可用生成函数表示:

$$G_0(x) = \sum_{k=1}^{\infty} p_k x^k,$$

且有  $G_0(0) = 1, z = \sum_k k p_k$  是平均度数.

在 P2P 网络中, 度数高低反映了节点的性能, 度数较高的节点一般具有较好的网络带宽, 在线时间较长且稳定, 处理速度快; 度数较低的节点网络处理能力较差, 且具有不稳定性. 分别称这两种节点为高度数节点和低度数节点.

**定义 8 (高度数节点与低度数节点).** 高度数节点定义为  $N_{hd} = \{Node | Deg(Node) \geq d_h\}$ , 低度数节点定义为  $N_{ld} = \{Node | Deg(Node) \leq d_l\}$ .

**定理 3.** 时间段  $t$ , 发起并完成  $|Q_i| > 1, |Q_i| < |N|$  条搜索, 资源占有率初始为  $r_i(0)$ , 并假定搜索过程中每条搜索的成功率为  $1 - (1 - r_i(0))^{K \cdot TTL}$ , 节点所接收到的转发消息平均数量为  $ARC(Node)$ , 则有,

$$ARC(Node) \geq \frac{E(O_i) \cdot d_h}{z \cdot |N|}, Node \in N_{hd} \tag{2-6}$$

$$ARC(Node) \leq \frac{E(O_i) \cdot d_l}{z \cdot |N|}, Node \in N_{ld} \tag{2-7}$$

证明: 由文献 [19] 及复杂网络知识可知, 随机选择一条边  $e$ , 沿着此条边到达的节点度数分布 (除去  $e$ )  $g_k$  与  $p_k$  不同, 其分布表示为

$$G_1(x) = \sum_{k=1}^{\infty} g_k = x \frac{G_0'(1)}{z}, g_k = \frac{(k+1)p_{k+1}}{\sum_k k p_k} = \frac{(k+1)p_{k+1}}{z}.$$

由于随机漫步的每一步转发都可看作均匀随机抽样, 转发到高度数节点的概率为  $P_h = \sum_{k=d_h-1}^{d_{max}} g_k$ , 转发到低度数节点的概率为  $P_l = \sum_{k=1}^{d_l-1} g_k$ .  $|Q_i|$  条搜索的所转发的节点数为  $E(O_i)$ , 由独立无关性可知,  $E(O_i)$  次转发至高度数节点的期望值为  $E(O_i) \cdot P_h$ , 转发至低度数节点的期望值为  $E(O_i) \cdot P_l$ . 因此, 当  $Node \in N_{hd}$  时, 可推得

$$ARC(Node) = \frac{E(O_i) \cdot P_h}{|N_{hd}|} = \frac{E(O_i) \cdot \sum_{k=d_h-1}^{d_{max}} (k+1)p_{k+1}}{|N| \cdot z \cdot \sum_{k=d_h-1}^{d_{max}} p_{k+1}} \geq \frac{E(O_i) \cdot d_h \cdot \sum_{k=d_h-1}^{d_{max}} p_{k+1}}{z \cdot |N| \cdot \sum_{k=d_h-1}^{d_{max}} p_{k+1}} = \frac{E(O_i) \cdot d_h}{z \cdot |N|}.$$

当  $Node \in N_{id}$  时,可推得

$$ARC(Node) = \frac{E(O_i) \cdot P_l}{N_{id}} = \frac{E(O_i) \cdot \sum_{k=1}^{d_i-1} (k+1) p_{k+1}}{|N| \cdot z \cdot \sum_{k=1}^{d_i-1} p_{k+1}} \leq \frac{E(O_i) \cdot d_i \cdot \sum_{k=1}^{d_i-1} p_{k+1}}{z \cdot |N| \cdot \sum_{k=1}^{d_i-1} p_{k+1}} = \frac{E(O_i) \cdot d_i}{z \cdot |N|} \quad \square$$

因此,节点度数越高,被转发的次数越多,从而证明了随机漫步易转向高度数节点.对于每条漫步,  $TTL$  时间内最终能转发到高度数节点的概率为  $1-(1-P_h)^{K \cdot TTL}$ . 由  $P_h$  的定义可知,  $P_h$  的大小与高度数节点的度数及所占的比率有关,若给定固定的比率,则显然,无标度网络中随机漫步最终转发到高度数节点的效果最为明显.高度数节点由于接收到的随机漫步消息比较多,可记录大量历史记录信息,并作为信息交换的枢纽,随机漫步搜索就能比较快速地获取到所需要的资源,这样,对于稀有资源,其成功率可近似为

$$1-(1-P_h)^{K \cdot TTL}.$$

### 2.4 逆向随机漫步搜索性质

逆向搜索的含义是资源主动寻找请求,与正向搜索不同,其成功率的定义有所区别:  $n$  条正向搜索找到同一个节点的资源可视为  $n$  次成功,而  $n$  条逆向搜索找到同一个节点的请求只能视为成功 1 次.本节对逆向随机漫步搜索(reverse random walk search,简称 RRWS)进行探讨.

**定理 4.** 当  $|Q_i| \geq |R_i| \geq 4$  时,  $R_i$  副本的逆向随机漫步搜索到不同请求的数量期望值  $E(Q_i)_b$  有

$$\frac{|R_i|}{3} (1-(1-q_i)^{K \cdot TTL}) + \frac{3}{2} < E(Q_i)_b < \frac{|R_i|}{3} (1-(1-q_i)^{K \cdot TTL}) + \frac{5}{2} \quad (2-8)$$

证明:问题相当于:对于一个整数  $n$ ,将其分解为多个整数的和的形式,求分解式中长度的期望值.

当  $n$  为奇数,  $n \geq 4$  时,所产生的分解式共有  $(n^2+3)/4$ ,其中,长度为 2 和 3 的分解式数目为  $(n-1)/2$ ,其概率为  $\left(\frac{n-1}{2}\right) / \left(\frac{n^2+3}{4}\right)$ ;长度为 4 和 5 的数目为  $(n-1)/2-2$ ,其概率为  $\left(\frac{n-1}{2}-2\right) / \left(\frac{n^2+3}{4}\right)$ ;以此类推,长度为  $n-1$  和  $n$  的数目为 1;加上  $n=n$  这个分解式,最后得到期望值为

$$\frac{2n^3+9n^2+22n-9}{6(n^2+3)} < \frac{n}{3} + \frac{3}{2} + \frac{11}{3n} < \frac{n}{3} + \frac{5}{2}, \quad \frac{2n^3+9n^2+22n-9}{6(n^2+3)} > \frac{2n^3+6n+9n^2+27}{6(n^2+3)} > \frac{n}{3} + \frac{3}{2}.$$

当  $n$  为偶数,  $n \geq 4$  时,同理可推得

$$\frac{2n^3+9n^2+22n}{6(n^2+4)} < \frac{n}{3} + \frac{3}{2} + \frac{11}{3n} < \frac{n}{3} + \frac{5}{2}, \quad \frac{2n^3+9n^2+22n}{6(n^2+4)} > \frac{2n^3+8n+9n^2+36}{6(n^2+4)} > \frac{n}{3} + \frac{3}{2}. \quad \square$$

由定理 4 可以确定  $E(Q_i)_b$  的范围,其近似值可认为是  $\frac{|R_i|}{3} (1-(1-q_i)^{K \cdot TTL})$ . 这说明,逆向搜索能够保证一定程度的成功,由  $1-(1-q_i)^{K \cdot TTL}$  可以看出,逆向搜索能够提高分发的速度,进而提高  $r_i$  变化的速度.

**定理 5.**  $|Q_i| \geq |R_i| \geq 4$ ,当正向搜索和逆向搜索的  $K \cdot TTL$  相等时

$$E(Q_i)_b - E(Q_i) < \frac{5}{2} \quad (2-9)$$

证明:设  $f(R_i) = \frac{|R_i|}{3} (1-(1-q_i)^m) + \frac{5}{2} - |Q_i| \left(1 - \left(1 - \frac{|R_i|}{|N|}\right)^m\right)$ , 有  $f''(R_i) = \frac{q_i m(m-1)}{|N|} \left(1 - \frac{R_i}{N}\right)^{m-2} > 0$ , 所以其函数图形上凹,且最大值出现在函数两个端点.而  $f(0) < \frac{5}{2}$ ,  $f(Q_i) = \frac{5}{2} - \frac{2|Q_i|}{3} (1-(1-q_i)^m)$ ,  $f(0) > f(Q_i)$ , 在  $0 \leq R_i \leq Q_i$  范围内,  $f(0)$  为最大值;对于  $4 \leq R_i \leq Q_i$ ,显然有  $\frac{5}{2} > f(R_i) > E(Q_i)_b - E(Q_i)$ .  $\square$

因此,逆向搜索本身比正向搜索不具有明显优势,若要提高逆向搜索的期望值,就必须增加  $K \times TTL$ ,这必然会带来过高的网络负担.

## 2.5 双向随机漫步搜索

**定义 9(双向随机漫步搜索, BRWS).** 随机漫步中  $S=\langle Q_i, m_i, F, H_i, O_i \rangle$ , 若节点集合  $Q_i$  表示发起请求此类资源的集合, 同时表示本身推送所携带资源;  $H_i$  表示拥有资源  $m_i$  的集合, 同时也表示需要资源的集合. 这种采用 piggybacks 方式获取资源的搜索过程, 称为双向随机漫步搜索.

在正向搜索中, 在同一条转发路径中同时进行逆向搜索, 将资源信息加入到转发消息中, 并与中途每一个经过的节点进行资源信息交换, 依次沿搜索路径迭代传递. 将逆向搜索嵌入到正向搜索中形成双向搜索. 其中, 资源信息包括节点所拥有的资源以及资源更改记录. 本文后续几节将对 BRWS 展开详细论述.

### 2.5.1 BRWS 数据结构和操作

**定义 10(BRWS 消息格式).** 双向搜索的消息格式统一定义为一个九元组:

$$Message = \langle ID, Type, QNode, ONode, FNode, Item, TTL, IRecord, CRecord \rangle,$$

其中,  $ID$  代表消息的唯一标识;  $Type$  代表消息类型 (forward, hit);  $QNode$  代表发起搜索资源的节点;  $ONode$  表示拥有资源的节点;  $FNode$  是转发此消息的节点, 即直接邻居节点;  $Item$  代表所搜索的资源;  $TTL$  表示消息的生存时间;  $IRecord$  表示所携带的资源拥有信息 (item record);  $CRecord$  是所携带的资源修改信息 (changed record).

$IRecord$  和  $CRecord$  是比较重要的数据结构, 其所记录的数据元素的格式分别为

$$IRecord.element = \langle Node, Item, Time \rangle, CRecord.element = \langle Node, Item, State, Time \rangle,$$

其中,  $IRecord$  中的  $Time$  表示资源产生的时间;  $CRecord$  中的  $Time$  表示资源修改的时间;  $State$  表示资源的状态 (Modify, Delete), 即修改和删除状态.

**定义 11(BRWS 节点记录).** 节点维护的记录定义为

$$NR = \langle ItemList, QueryList, Neighbors, IRecord, CRecord \rangle.$$

$ItemList$  是所拥有的资源列表, 数据元素的格式为  $ItemList.element = \langle Item \rangle$ ;  $QueryList$  为节点的请求资源记录列表, 数据元素的格式为  $QueryList.element = \langle Item, Time, Hit, NH \rangle$ , 其中,  $Time$  为发起搜索的时间,  $Hit$  表示是否命中, 为布尔值,  $NH$  表示命中的次数;  $Neighbors$  是邻居节点列表, 数据元素的格式为  $Neighbors.element = \langle Node \rangle$ ;  $IRecord$  和  $CRecord$  分别记录所获知的资源拥有信息和资源修改信息,  $NR$  和  $Message$  都维护着这两种数据结构.

$IRecord$  和  $CRecord$  所记录的信息数量均有上限, 在  $NR$  和  $Message$  中的大小不一, 节点的记录上限  $MaxNIC$  要大于消息中记录的上限  $MaxMIC$ :

$$\begin{cases} MaxNIC = MaxSize(NR.IRecord) = MaxSize(NR.CRecord) \\ MaxMIC = MaxSize(Message.IRecord) = MaxSize(Message.CRecord). \\ MaxNIC \geq MaxMIC \end{cases}$$

$IRecord$  和  $CRecord$  的记录数据均按照 FIFO 方式排列, 采用队列结构存储.

**定义 12(BRWS 节点转发策略).** 若节点  $IRecord$  中有关于资源  $Item$  的记录, 由与这些记录相关的节点组成的集合记为  $Nodes(Item\_NRIR)$ , 有

$$Nodes(Item\_NRIR) = \{ Node | \exists \langle Node, Item, Time \rangle \in NR.IRecord \}.$$

当此节点转发搜索资源  $Item$  消息时, 若存在非空集合  $Nodes(Item\_NRIR)$ , 则从此集合中随机选择一个节点进行转发; 若不存在这样的非空集合, 则从其邻居列表中  $Neighbors$  随机选择一个节点进行转发, 即采用均匀分布概率选择节点, 选择节点的概率设为  $p(Node)$ , 节点转发策略可形式化描述为

$$DestNode = \begin{cases} Node \in Nodes(Item\_NRIR) \wedge p(Node) = \frac{1}{|Nodes(Item\_NRIR)|}, & \text{if } \exists \langle Node, Item, Time \rangle \in NR.IRecord \\ Node \in Neighbors \wedge p(Node) = \frac{1}{|Neighbors|}, & \text{(if } \exists \langle Node, Item, Time \rangle \in IRecord) NR.IRecord \end{cases} \quad (2-10)$$

**定义 13(BRWS 记录更新策略).** 节点  $Node_s$  收到消息时, 需要依次进行 3 方面的更新操作, 首先给出几个发

起搜索的节点只执行 Update 3 的操作.

**Update 1.** 收到消息时,  $Node_s$  根据  $Message$  的  $IRecord$  和  $CRecord$  来更新本身的记录. 给定几个更新条件:

$$\exists \langle Node, Item, Time_i \rangle \in NR.IRecord \wedge \exists \langle Node, Item, Time_j \rangle \in Message.IRecord \quad (\text{条件 1})$$

$$\exists \langle Node, Item, Time_i \rangle \in NR.IRecord \wedge \exists \langle Node, Item, Time_j \rangle \in Message.IRecord \quad (\text{条件 2})$$

$$\exists \langle Node, Item, Stae, Time_i \rangle \in NR.CRecord \wedge \exists \langle Node, Item, Stae, Time_j \rangle \in Message.CRecord \quad (\text{条件 3})$$

$$\exists \langle Node, Item, Stae, Time_i \rangle \in NR.CRecord \wedge \exists \langle Node, Item, Stae, Time_j \rangle \in Message.CRecord \quad (\text{条件 4})$$

首先, 扫描  $Message$  中  $IRecord$  的每一条记录, 若满足条件 1, 则追加记录  $\langle Node, Item, Time_j \rangle$  到  $Node_s$  的  $NR.IRecord$  的队尾; 如果满足条件 2, 则比较  $Time_i$  和  $Time_j$ , 若  $Time_j > Time_i$ , 则  $Time_j$  替换  $Time_i$ , 并将其位置移动到  $NR.IRecord$  队尾, 否则放弃.

然后, 扫描  $Message$  中  $CRecord$  的每一条记录, 若满足条件 3, 则追加  $\langle Node, Item, Stae, Time_i \rangle$  到节点  $NR.CRecord$  队尾; 若满足条件 4, 则比较  $Time_i$  和  $Time_j$ , 如果  $Time_j > Time_i$ , 则  $Time_j$  替换  $Time_i$ , 并将其位置移动到  $NR.CRecord$  队尾. 接着扫描  $NR.IRecord$ , 若  $\exists \langle Node, Item, Time_i \rangle \in NR.IRecord$ , 当  $Stae$  为  $Modify$  时, 则用  $Time_j$  替换  $Time_i$ , 并将其位置移动到  $NR.IRecord$  队尾; 如果为  $Delete$ , 则需要在  $NR.IRecord$  删除此条记录.

记录更新过程中, 如果  $NR.IRecord$  和  $NR.CRecord$  容量达到上限, 则利用 FIFO 方法进行替换记录.

**Update 2.** 当根据定义 12 的节点转发策略获得待转发节点后, 则需要根据当前的  $NR.IRecord$  和  $NR.CRecord$  来对  $Message$  的  $IRecord$  和  $CRecord$  信息进行更新, 其方法与 Update 1 类似, 替换策略依然采用 FIFO.

**Update 3.** 当完成上述操作后,  $Message.IRecord$  根据  $Node_s$  的  $NR.ItemList$  更新自身记录, 当前的时间为  $Now$ , 给定如下两个条件:

$$\exists \langle Node_s, Item, Time_i \rangle \in Message.IRecord \wedge \exists Item \in NR.ItemList \quad (\text{条件 5})$$

$$\exists \langle Node_s, Item, Time_i \rangle \in Message.IRecord \wedge \exists Item \in NR.ItemList \quad (\text{条件 6})$$

对于每一个  $Item$ , 若满足条件 5, 则将  $\langle Node_s, Item, Now \rangle$  追加到  $Message.IRecord$  队尾. 如果满足条件 6, 则将  $Now$  替换掉  $Time_i$ , 并将其位置移动到  $Message.IRecord$  队尾. 容量若达到上限, 则利用 FIFO 方法进行替换记录.

### 2.5.2 搜索算法

- (1)  $QueryNode$  发起对资源  $Item_i$  的搜索, 并执行定义 13 中的 Update 3 操作;
- (2) 网络中任何节点  $Node_k$  对自身所拥有的资源修改或删除操作都记录在该节点的  $NR.CRecord$  中, 给定如下两个条件:

$$\exists \langle Node_k, Item, State, Time_i \rangle \in NR.CRecord \wedge \exists Item \in NR.ItemList \quad (\text{条件 7})$$

$$\exists \langle Node_k, Item, State, Time_i \rangle \in NR.CRecord \wedge \exists Item \in NR.ItemList \quad (\text{条件 8})$$

在时刻  $MTime$  对自身拥有的资源  $Item$  发生修改时, 若满足条件 7, 则将  $\langle Node_k, Item, Modify, MTime \rangle$  追加到  $NR.CRecord$  队尾; 若满足条件 8, 则用  $MTime$  替换  $Time_i$ , 并将对应记录移动到队尾. 当在  $DTime$  时刻对自身拥有的资源  $Item$  发生删除时, 若满足条件 7, 则将  $\langle Node_k, Item, Delete, DTime \rangle$  追加到  $NR.CRecord$  队尾; 若满足条件 8, 则用  $DTime$  替换  $Time_i$ , 并将对应记录移动到队尾, 然后在  $NR.ItemList$  中删除  $Item$  记录;

- (3)  $Node_k$  收到消息时, 无论是转发消息还是命中消息类型, 首先都执行定义 13 中的 Update 1 和 Update 2 操作, 即根据  $Message$  的  $IRecord$  和  $CRecord$  来更新本身的记录;
- (4)  $Node_k$  收到转发消息时, 即  $Message.Type = forward$ , 若有

$$\exists \langle Item, Time_i, false, 0 \rangle \in NR.QueryList \wedge \exists \langle Node, Item, Time_i \rangle \in Message.IRecord,$$

设  $Message.IRecord$  中与该  $Item$  相关的节点组成的集合记为  $Nodes(Item\_MsIr)$ , 即

$$Nodes(Item\_MsIr) = \{Node \mid \exists \langle Node, Item, Time \rangle \in Message.IRecord\},$$

则从  $Nodes(Item\_MsIr)$  中随机选择一个节点进行查询; 如果查询失败, 则继续从  $Nodes(Item\_MsIr)$  选择其他节点进行查询, 并执行步骤(2)中删除状态的更新操作, 直到查询成功或者没有节点可以选择,



然后转向步骤(5);

- (5)  $Node_k$  收到转发消息时,若 $\exists Item \in NR.ItemList \wedge Item = Message.Item$ ,则向  $QueryNode$  发起命中信息;否则,转向步骤(6);
- (6)  $Message.TTL$  自动减 1,如果此时  $Message.TTL$  变为 0,则转发终止;否则,转向步骤(7);
- (7) 按照定义 12 的节点转发策略来选择节点,首先探测节点是否掉线,如果候选节点已掉线,则执行步骤(2)中删除状态的更新操作,然后重新按照定义 12 选择节点,直到没有节点可以选择,终止操作;否则,执行定义 13 中的 Update 3,将消息转发至候选节点.

## 2.6 性能分析

假定同时发起  $S+1$  个对不同资源的请求,第  $S+1$  个请求为  $Q_i, K \cdot TTL=l$ ,其他每个请求在搜索过程中遇到资源  $m_i$  信息的概率为  $1-(1-r_i)^l$ ,则携带  $m_i$  信息的请求个数期望值为  $W=S \cdot (1-(1-r_i)^l)$ , $Q_i$  被其他携带  $m_i$  信息的请求经过的概率为  $1-\left(1-\frac{1}{n}\right)^{l \cdot W}$ . 在 BRWS 搜索过程中,对资源  $m_i$  的信息会沿路径传递下去,所以  $S$  个请求将资源  $m_i$  的

比率提高到  $r'_i = \frac{\sum_{j=1}^S E(O_j)}{2 \cdot |N|} + r_i$ . 由此可知, $Q_i$  的成功率和网络开销分别为

$$SR_i = 1 - \left(1 - \frac{1}{|N|}\right)^{l \cdot W} \cdot (1-r_i)^l \quad (2-11)$$

$$E(O_i) \leq \frac{1-(1-r_i)^l}{r_i} \quad (2-12)$$

如果考虑 Random Walk 易转向高度数节点的特点,大部分资源的信息被记录在高度数节点中,高度数节点接收到的携带  $m_i$  的转发信息大约为  $W \cdot P_h$ ,则平均每个高度数节点拥有信息  $m_i$  的平均数为  $\frac{W \cdot d_h}{z \cdot |N|}$ . 显然,只要网络中存在充分高的度数,那么,随着搜索数量的增加,每种请求都至少会在高度数节点上找到需要的资源.很明显, BRWS 能够有效改善稀有资源的搜索.实际网络中,节点的不断加入和退出、节点上资源的不断删除带来了扰动现象,一旦资源发生变化,所指向这个资源的索引将会失效.为此,在 BRWS 搜索中除携带资源信息以外,还需要携带资源的变化信息,逆向搜索同时负责了资源信息的更新.假设指向节点  $Node$  删除了资源  $m_i$ ,BRWS 发现资源发生变化,就会沿着路径传递下去,假定依次发起  $S$  个请求,经过  $Node$  的搜索为  $\frac{S \cdot K}{|N|}$ ,假设  $S$  个请求的平均搜索路径为  $m$ ,显然,资源变化更新的速度为

$$v \approx \left(\frac{m}{2}\right)^{\frac{S \cdot K}{|N|}} \quad (2-13)$$

网络中请求数量越多,更新速度越快.由于 Random Walk 易转向高度数节点,则在这些节点上关于资源的信息会很快被更新到,从而减少了搜索的失败率.

由于对  $IRecord$  和  $CRecord$  进行了限制,总的空间开销最坏情况下为  $O(2 \cdot \text{MaxNIC} \cdot |N|)$ ,对于每条搜索的开销最高为  $O(2 \cdot \text{MaxMIC} \cdot l)$ .

## 3 模拟实验与对比分析

本实验通过扩展 Peersim 代码进行模拟,所涉及的网络拓扑包括 Random Graph,Scale Free 网络,Small World 网络.实验由 5 部分组成,比较的方法包含 APS,Random Walk,P2PBSN 和 PQR.

- (1) 静态网络环境下搜索成功率、搜索延迟和网络开销对比;
- (2) 动态网络环境下的搜索成功率、搜索延迟和网络开销对比;
- (3) 动态网络环境下稀有资源搜索成功率对比;

- (4) 动态环境下存储开销对比分析;
- (5) 路由转发表的可靠性分析.

资源和请求均服从 Zipf 分布,请求的发起时间采用均匀随机分布,实验默认参数见表 1.

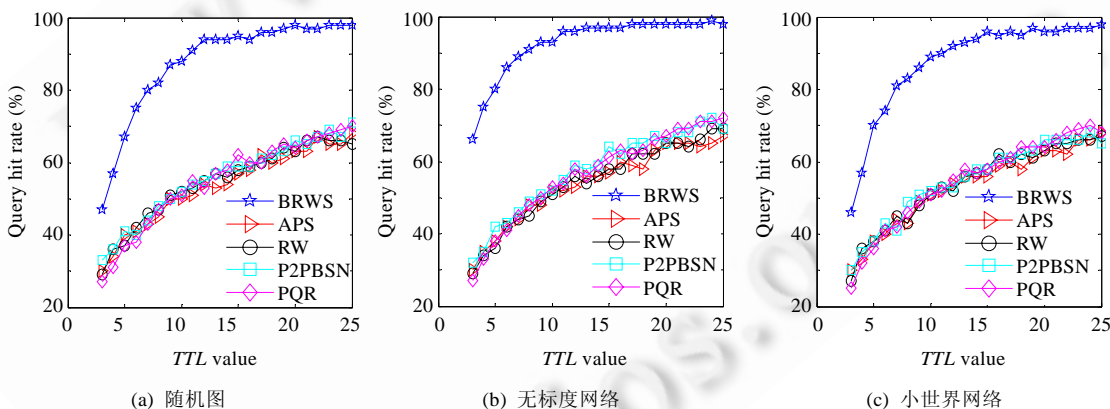
**Table 1** Simulation parameters and default values

**表 1** 模拟参数及默认值

Simulation parameters	Default values
Network size	10 000
Number of query request	1 000
Number of object	100
Object replication ratio	2%
Replication distribution	Zipf ( $\alpha=0.82$ )
Query distribution	Zipf ( $\alpha=0.9$ )
Average node degree in every network	20
Walkers deployed (k)	3
$\beta$ in small world network	0.5
$\alpha$ in PQR	0.5
$\lambda$ in PQR	2
C in PQR	10
$\alpha$ in P2PBSN	6
$\beta$ in P2PBSN	-1
MaxNIC in BRWS	100
MaxMIC in BRWS	20

**3.1 静态网络环境下性能对比**

Walkers deployed (K)取固定值 3,TTL 的范围是 3~25.图 1 给出在 3 种拓扑中的搜索成功率,由图 1 可以看出,BRWS 相对于其他 4 种方法提高了 30%左右;在 Scale Free 网络中,即使在 TTL 较小的情况下,成功率也达到了 60.证明,随机漫步在 Scale Free 网络中具有转发至高度数节点概率比较高的特点.其他传统方法搜索命中率差不多,其原因有可能与搜索数量过少有关系.



**Fig.1** Query hit rate vs. TTL value in static environment

图 1 静态网络环境下搜索成功率比较

本文定义搜索延迟是面向用户角度、第 1 条收到的回应消息所用的跳数,如图 2 所示.随着 TTL 的增加,BTWS 逐步稳定在 TTL=3,而其他搜索方法则呈阶梯状逐步上升.

图 3 是每条搜索产生的平均消息数,BRWS 的平均消息数量远远小于其他几种拓扑.而在 Scale Free 网络中,平均消息数更小,这是由于,除了逆向搜索本身查找到请求以外,大部分搜索命中都是在信息量比较大的高度数节点上完成的.

在静态环境下,BRWS 的 Message 中只有 IRecord 记录,其在 3 种网络拓扑中的平均记录数如图 4 所示.其结

果表明,随着 *TTL* 的增加,记录数会逐渐趋于稳定.正是由于搜索成功率的提高,从而导致经过较短的跳数就可以找到所需要的资源.

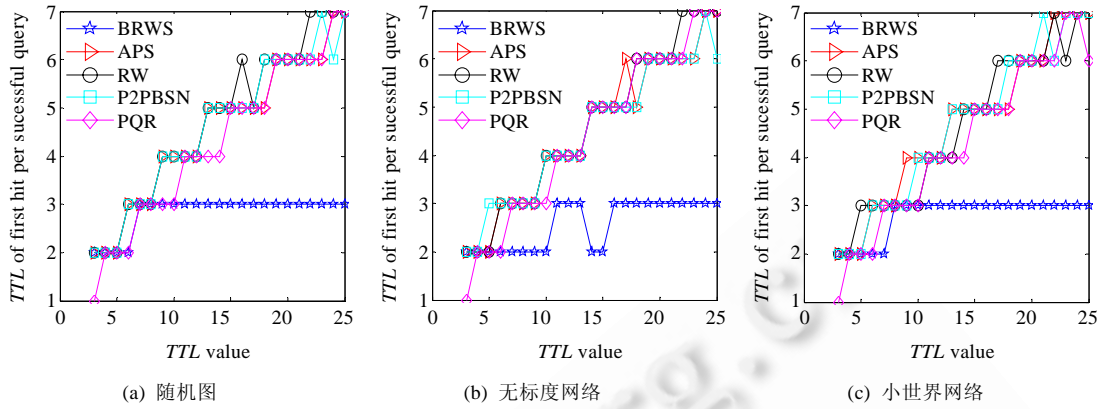


Fig.2 Average hops of the first hit per query in static environment

图 2 静态网络环境下搜索延迟对比

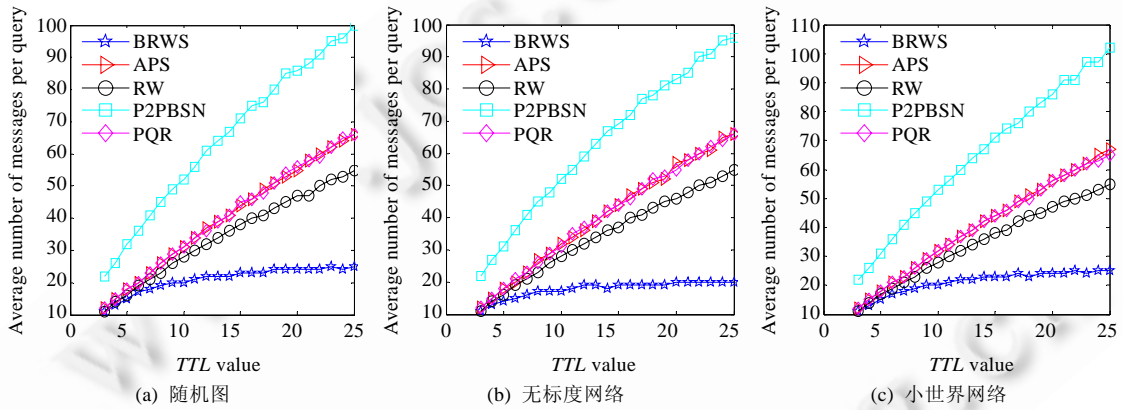


Fig.3 Average number of messages per query vs. *TTL* in static environment

图 3 静态网络环境下平均每条搜索消息数比较

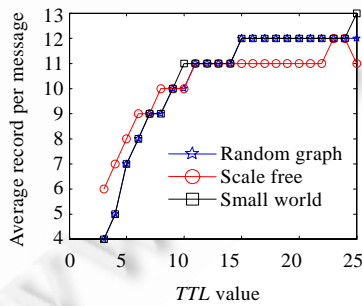


Fig.4 Average number of record per message in static environment

图 4 静态网络环境下 BRWS 平均每条消息记录数

网络总开销需要对比平均每条搜索产生的消息的网络开销,对于后者,本文指的是平均每条搜索所产生的

IP 数据包的字节数.由于每条消息的总的字节数都较小,所以按照不产生分片来计算,其在静态环境下,平均每条搜索产生的字节数如图 5 所示.

可以看出,Random Walk 平均每条搜索的网络总开销最小,BRWS 略高于 Random Walk;但随着 TTL 的增加而逐渐趋于稳定,并与 Random Walk 的差距缩小.APS 和 PQR 基本开销一致,随着 TTL 的增加,基本呈线性增长,但增长速度较慢;而 P2PBSN 的增长速度最快,最高超过了 20K.

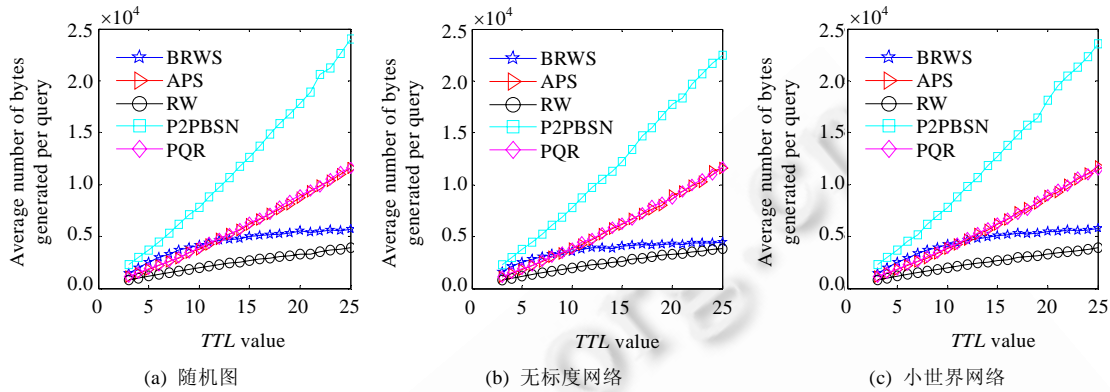


Fig.5 Average number of bytes per query in static environment

图 5 静态网络环境下平均每条搜索的网络总开销比较

### 3.2 动态网络环境下性能对比

本文模拟动态网络环境采取的策略是,保持网络节点数不变,将模拟时间分成 50 个周期,每个周期随机选取 200 个节点,随机删除每个节点上的一种资源,增加 10 个节点,并按 Zipf 分布分配资源.图 6 是动态网络环境下的搜索成功率,由于 BRWS 分发速度比较快,并且携带了资源修改信息,可以快速更新已删除的资源信息,因此,当  $TTL \geq 10$  时,与静态网络环境成功率差别很小,相对其他几种搜索成功率有明显提高.APS,PQR 和 P2PBSN 比 Random Walk 更能适应动态网络环境,成功率明显较高;并且相对于另外两种网络,PQR 和 P2PBSN 在 Scale Free 网络中的搜索成功率要比 APS 明显偏高.

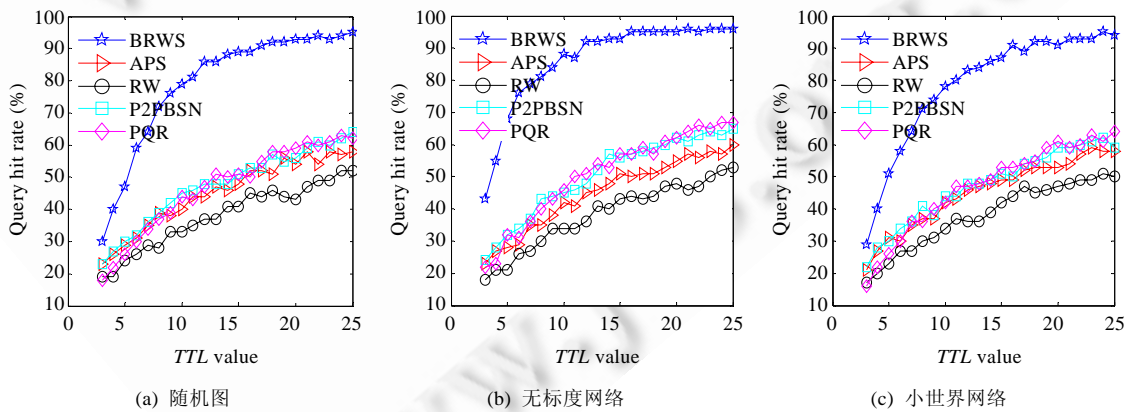


Fig.6 Query hit rate vs. TTL value in dynamic environment

图 6 动态网络环境下搜索成功率比较

动态环境下的搜索延迟,BRWS 在 Random Graph 和 Small World 中逐渐趋于 4,而在 Scale Free 网络中基本上稳定在 3,其他几种都平均增加了两个跳数(如图 7 所示).

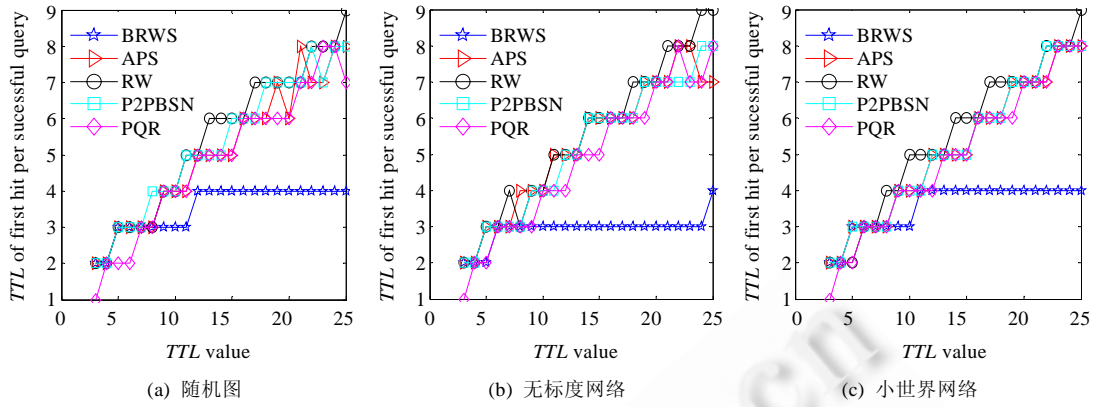


Fig.7 Average hops of the first hit per query in dynamic environment

图 7 动态网络环境搜索延迟对比

图 8 是动态网络环境下的网络开销,周期性地随机删除服从 Zipf 分布的资源,初始分布中副本数量最多的资源比重下降较快,导致 BRWS 转发的路径会比静态网络环境下的偏长.因此,平均消息数大约比静态网络中增加了 5~10 个消息数,但仍远远低于其他几种.

与静态网络环境一样,APS 和 PQR 略高于 Random Walk,P2PBSN 仍保持最高的平均消息数,几乎增加了 10%~20%.

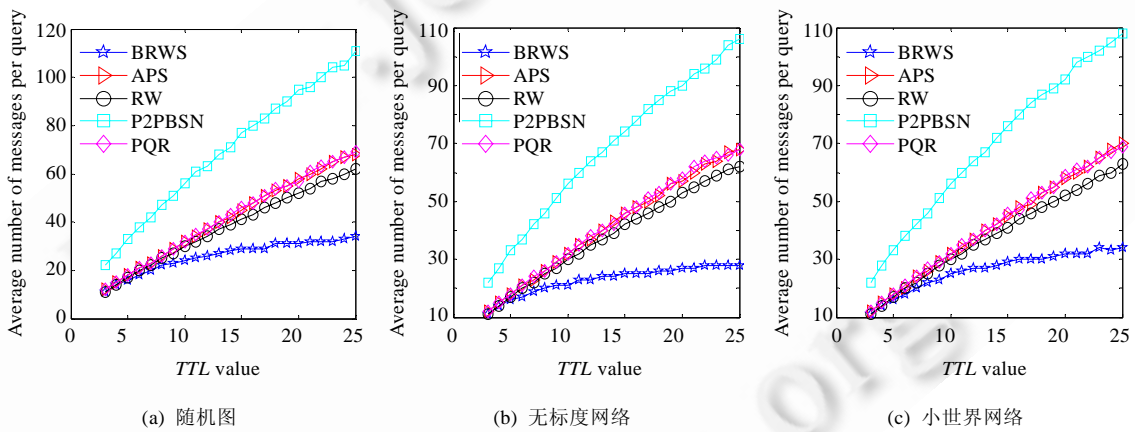


Fig.8 Average number of messages per query vs. TTL in dynamic environment

图 8 动态网络环境下平均每条搜索消息数比较

在动态环境下,BRWS 的 Message 中包含 IRecord 和 CRecord 记录,其在 3 种网络拓扑中的平均记录数如图 9 所示,IRecord 记录数量基本与在静态环境下相比变化不大,IRecord 和 CRecord 记录之和基本上趋于在 TTL 值附近.从这个角度来看,网络开销会与发起搜索数量成正比.

如图 10 所示,动态环境下其他几种方法的变化不大.而由于 CRecord 记录的增加,BRWS 网络总开销相对于静态环境有所提高,但最终会趋于与 ASP,PQR 的开销相一致,大约为 Random Walk 的 2~3 倍左右.

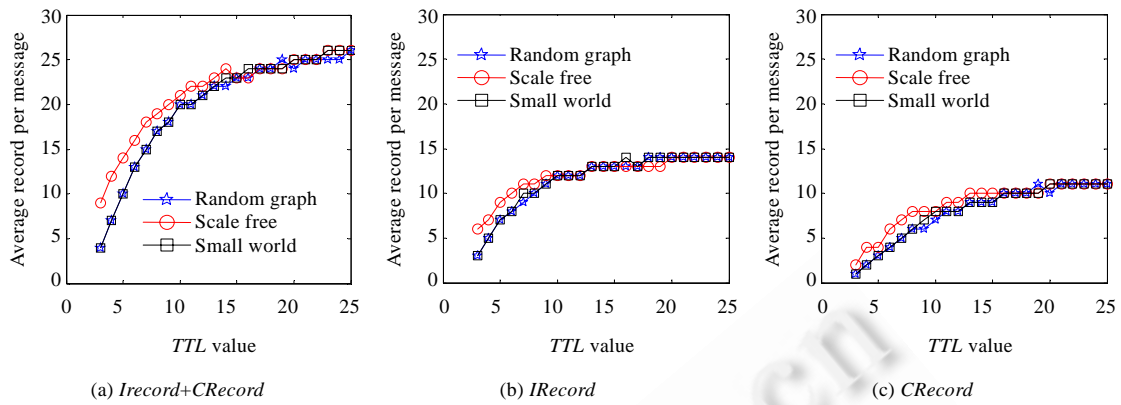


Fig.9 Average number of record per message in dynamic environment

图 9 动态网络环境下 BRWS 平均每条消息记录数

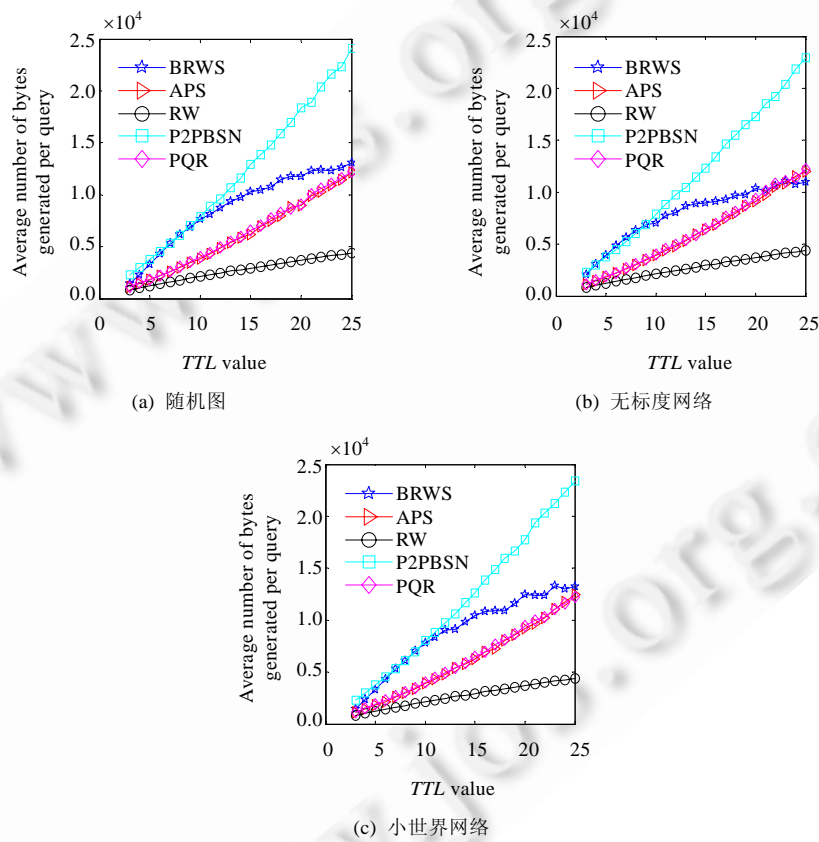


Fig.10 Average number of bytes per query in dynamic environment

图 10 动态网络环境下平均每条搜索的网络总开销比较

### 3.3 稀有资源搜索成功率对比

资源副本的分布服从 Zipf 分布,因此本实验将 100 种资源的后 10 种作为稀有资源,考察静态和动态网络环境下,这 10 种资源请求的成功率。

如图 11 和图 12 所示, BRWS 随着  $TTL$  的增加, 成功率逐渐增高, 而其余几种基本上保持在较低的成功率水平, 没有变化, 说明 BRWS 能够较好地支持稀有资源搜索. 在 Scale Free 网络中, 成功率升高得比较快, 且趋于稳定, 这是因为 Scale Free 网络度节点跨度大, 稀有资源的搜索容易在高度数节点上查到所需要的资源信息.

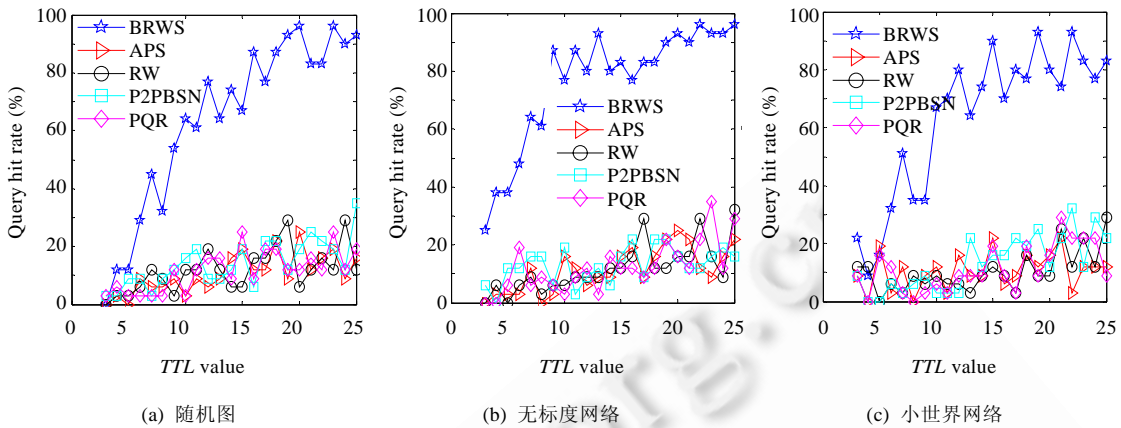


Fig.11 Query hit rate of rare items vs.  $TTL$  value in static environment

图 11 静态网络环境下稀有资源搜索成功率比较

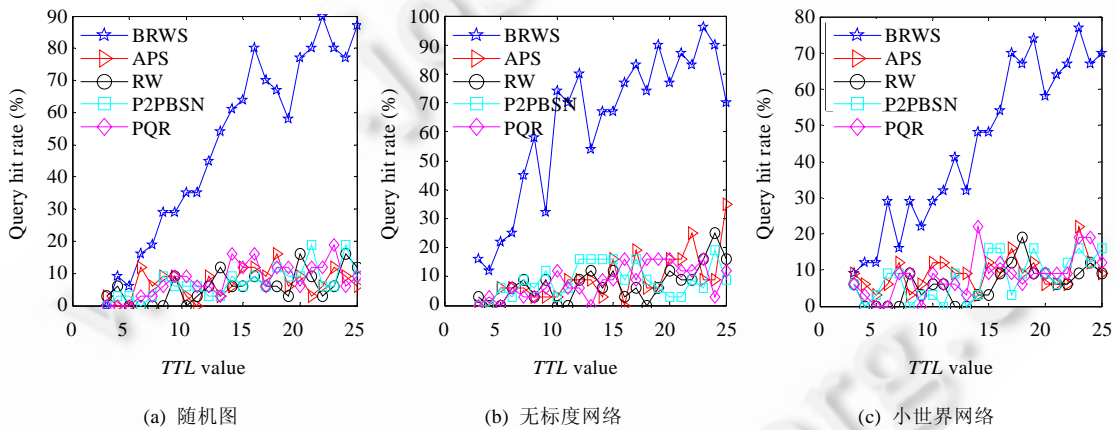


Fig.12 Query hit rate of rare items vs.  $TTL$  value in dynamic environment

图 12 动态网络环境下稀有资源搜索成功率比较

### 3.4 存储开销对比分析

为了分析存储开销, 将模拟时间平均分成 100 个时间段, 观察每个时间段平均每个节点的存储开销, 与网络总开销一样, 以字节为单位. 从图 13 可以看出, PQR 和 APS 的开销都较小, P2PBSN 远远大于其他几种网络, 并且在 Scale Free 网络中所消耗的存储开销比其他两种网络要多 2 000 多字节. BRWS 开销较大, 但最高维持在 600 字节左右, 在实际系统中, 这种开销是微乎其微的. 而本实验所允许的阈值不会超过 1 200 字节.

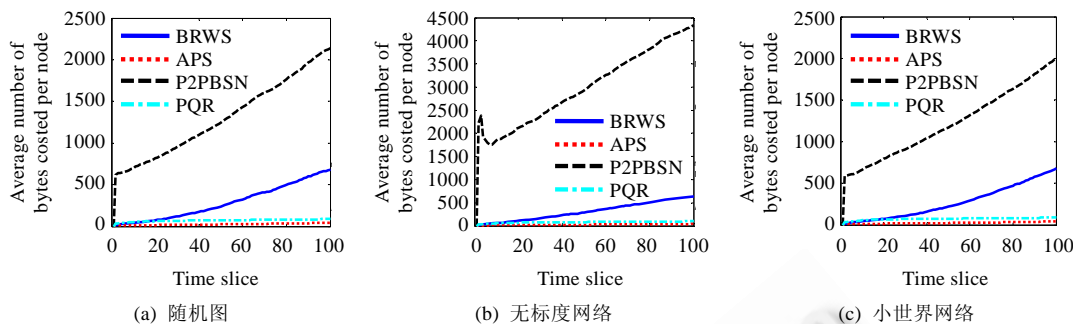


Fig.13 Storage overhead in dynamic environment

图 13 动态网络环境下存储开销对比

3.5 路由转发表可靠性分析

基于历史命中记录建立的转发表由于需要时间和搜索量的积累,因此 P2P 的扰动现象使其可靠性比较低.本实验对比 BRWS 与 APS,PQR 和 Random Walk 的路由转发表可靠性,为了能够积累有效的路由转发表,特将搜索数量增至 10 000,仍采用 Zipf 分布.由于 P2PBSN 随搜索数量的增大,网络开销急剧增加,因此本实验部分没有予以对比.将模拟时间平均分成两段,在静态环境下,两部分均未加入扰动因素;而在动态网络环境下,前半部在后半部分时间段内加入扰动因素,扰动周期为模拟时间的 0.1%,在扰动时刻,最多可随机删除 200 个资源副本.实际上,前半部分相当于建立转发表的过程,大约 5 000 左右的搜索数量足以建立有效的路由表.后半部分的扰动是为了验证前半部分时间的可靠性,对比方法是比较不加入扰动因素的静态环境下,后半部分命中率的下降幅度.

为了对比明显,将 BRWS 的 TTL 设置为 6,MAXNIC 和 MAXMIC 均设置为 5.其余 3 种方法的 TTL 设置为 15.从图 14 可以看出,即便在这种有差别的情况下,BRWS 的搜索命中率仍高出其他方法许多;并且在动态环境下,后半部分的时间搜索命中率下降幅度较小,大约 5%左右,而搜索命中率却仍高于 80%.

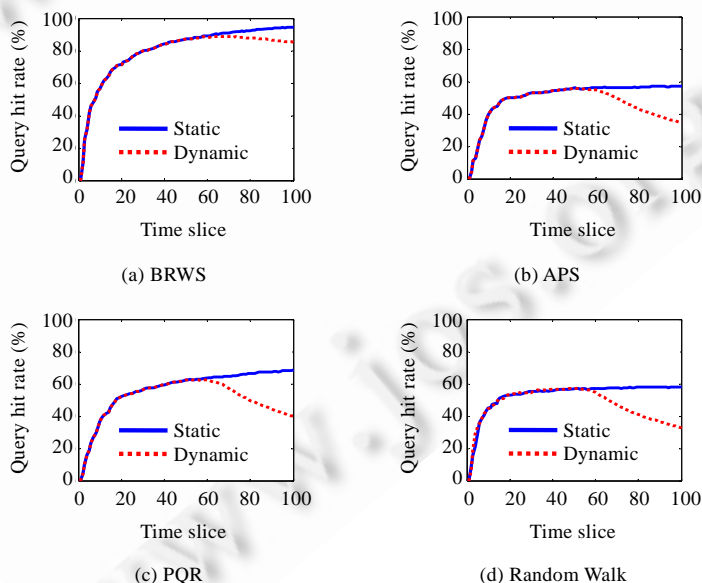


Fig.14 Trend of query hit rate in static and dynamic environments

图 14 静态和动态网络环境下命中率趋势

PQR 在静态网络环境下,随着路由转发表的完善,搜索命中率明显高于 Random Walk;但 APS 改善得不明



显,其原因是由转发表中的初始索引值无法确定和规范化所致,过高和过低的默认索引值会造成转发概率均匀化. PQR 和 APS 的转发表可靠性都较低,随着时间的推移,基本上都趋于与 Random Walk 相一致,下降幅度都超过了 20%,搜索命中率基本降低到 40% 以下.

#### 4 结束语

本文针对以往 Random Walk 改进算法的不足,提出了一种双向随机漫步搜索机制.首先,分析并从理论上证明了随机漫步易转向高度数节点的特点,单纯的逆向搜索会引发过高的网络开销.本文从理论分析入手,指出了这种不足,并提出根据随机漫步的网络特性,将逆向搜索嵌入到正向搜索中,形成双向搜索 BRWS.同时,考虑到实际 P2P 网络存在动态变化这一情况,在 BRWS 中融入了抗扰动策略.本文从理论上证明,BRWS 能够提高包括稀有资源搜索在内的搜索成功率、降低网络开销、抗扰动能力强,给出了具体的数据结构和搜索流程.最后,通过模拟实验验证了 BRWS 能够以较小的网络开销,极大地提高搜索成功率.本文的不足之处是,对网络扰动的模拟比较单一,未来将重点研究各种动态环境下的搜索性能,并将考虑当网络不存在 Giant Component 的情况下如何进行搜索.

#### References:

- [1] Zhao BY, Kubiawicz JD, Joseph AD. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report, CSD-01-1141, Berkeley: University of California at Berkeley, 2001. 1–27.
- [2] Stoica I, Morris R, Karger D, Kaashoek MF, Balakrishnan H. Chord: A scalable peer-to-peer lookup service for Internet applications. ACM SIGCOMM Computer Communication Review, 2001,31(4):149–160. [doi: 10.1145/964723.383071]
- [3] Druschel P, Rowstron A. PAST: A large-scale, persistent peer-to-peer storage utility. In: Williams AD, ed. Proc. of the 8th Workshop on Hot Topics in Operating Systems. Elmau: IEEE Computer Society, 2001. 75–80. [doi: 10.1109/HOTOS.2001.990064]
- [4] Rowstron A, Druschel P. Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility. ACM SIGOPS Operating Systems Review, 2001,35(5):188–201. [doi: 10.1145/502059.502053]
- [5] Malkhi D, Naor M, Ratajczak D. Viceroy: A scalable and dynamic emulation of the butterfly. In: Proc. of the Principles of Distributed Computing. Monterey: ACM Press, 2002. 183–192. [doi: 10.1145/571825.571857]
- [6] Yang B, Garcia-Molina H. Improving search in peer-to-peer networks. In: Rodrigues LET, ed. Proc. of the 22nd Int'l Conf. on Distributed Computing Systems. Vienna: IEEE Computer Society, 2002. 5–14. [doi: 10.1109/ICDCS.2002.1022237]
- [7] Lü Q, Cao P, Cohen E, Li K, Shenker S. Search and replication in unstructured peer-to-peer networks. In: Proc. of the ACM SIGMETRICS, 2002 Int'l Conf. on Measurement and Modeling of Computer Systems. Marina Del Rey: Association for Computing Machinery, 2002. 258–259. [doi: 10.1145/514191.514206]
- [8] Kalogeraki V, Gunopulos D, Zeinalipour-Yazti D. A local search mechanism for peer-to-peer networks. In: Proc. of the Information and Knowledge Management. McLean: ACM Press, 2002. 300–307. [doi: 10.1145/584792.584842]
- [9] Gkantsidis C, Mihail M, Saberi A. Random walks in peer-to-peer networks. In: Proc. of the 23rd Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2004). Hong Kong: IEEE Computer Society, 2004. 130. [doi: 10.1109/INFOCOM.2004.1354487]
- [10] Tsumakos D, Roussopoulos N. Adaptive probabilistic search for peer-to-peer networks. In: Shahmehri N, ed. Proc. of the 3rd Int'l Conf. on Peer-to-Peer Computing (P2P 2003). Linköping: IEEE Computer Society, 2003. 102–109. [doi: 10.1109/PTP.2003.1231509]
- [11] Zheng QB, Lu XC, Zhu PD, Peng W. An efficient random walks based approach to reducing file locating delay in unstructured P2P network. In: Proc. of the Global Telecommunications Conf. (GLOBECOM 2005). New York: IEEE Communication Society, 2005. 5. [doi: 10.1109/GLOCOM.2005.1577783]
- [12] Ronasi K, Firooz M H, Pakravan M R, Avanaki AN. An enhanced random-walk method for content locating in P2P networks. In: Proc. of the 27th Int'l Conf. on Distributed Computing Systems WORKSHOPS. Toronto: IEEE Computer Society, 2007. 21. [doi: 10.1109/ICDCSW.2007.21]

- [13] Filali I, Huet F. Dynamic *TTL*-based search in unstructured peer-to-peer networks. In: Parashar M, ed. Proc. of the 10th IEEE/ACM Int'l Conf. on Cluster, Cloud, and Grid Computing (CCGrid 2010). Melbourne: IEEE Computer Society, 2010. 438–447. [doi: 10.1109/CCGRID.2010.66]
- [14] Zhang YM, Lu XC, Zheng QB, Li DS. An efficient search algorithm for large-scale P2P systems. Journal of Software, 2008,19(6): 1473–1480 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/19/1473.htm> [doi: 10.3724/SP.J.1001.2008.01473]
- [15] Huang YS, Meng XW, Zhang YJ. Strategy of content location of P2P based on the social network. Journal of Software, 2010, 21(10):2622–2630 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3647.htm> [doi: 10.3724/SP.J.1001.2010.03647]
- [16] Xu M, Zhou SG, Guan JH, Hu XH. A path-traceable query routing mechanism for search in unstructured peer-to-peer networks. Journal of Network and Computer Applications, 2010,33(2):115–127. [doi: 10.1016/j.jnca.2009.11.003]
- [17] Kumar A, Xu J, Zegura EW. Efficient and scalable query routing for unstructured peer-to-peer networks. In: Makki K, ed. Proc. of the 24th Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2005). Miami: IEEE Computer Society, 2005. 1162–1173. [doi: 10.1109/INFOCOM.2005.1498343]
- [18] Simonton E, Byung KC, Seidel S. Using gossip for dynamic resource discovery. In: Feng W, ed. Proc. of the 2006 Int'l Conf. on Parallel Processing (ICPP 2006). Columbus: IEEE Computer Society, 2006. 319–328. [doi: 10.1109/ICPP.2006.72]
- [19] Newman MEJ, Strogatz SH, Watts DJ. Random graphs with arbitrary degree distributions and their applications. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics, 2001,64(2 II):261181–261187. [doi: 10.1103/PhysRevE.64.026118]

#### 附中文参考文献:

- [14] 张一鸣,卢锡城,郑倩冰,李东升.一种面向大规模 P2P 系统的快速搜索算法.软件学报,2008,19(6):1473–1480. <http://www.jos.org.cn/1000-9825/19/1473.htm> [doi: 10.3724/SP.J.1001.2008.01473]
- [15] 黄永生,孟祥武,张玉洁.基于社会网络特征的 P2P 内容定位策略.软件学报,2010,21(10):2622–2630. <http://www.jos.org.cn/1000-9825/3647.htm> [doi: 10.3724/SP.J.1001.2010.03647]



马文明(1982—),男,山东德州人,博士生,主要研究领域为 P2P 内容分发网络,个性化推荐服务。



张玉洁(1969—),女,讲师,主要研究领域为网络服务,数字媒体。



孟祥武(1966—),男,博士,教授,博士生导师,主要研究领域为网络服务,内容分发,语义 Web。