软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(3):640−662 [doi: 10.13328/j.cnki.jos.004595]

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

面向用户需求的非结构化P2P资源定位泛洪策略*

何 明[1,2], 张玉洁[1,2], 孟祥武[1,2]

[1](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)

[2](北京邮电大学 计算机学院,北京 100876)

通讯作者: 何明, E-mail: heming0405@163.com

摘 要: 在非结构化 P2P 网络中,如何对用户所需资源进行快速、准确定位是当前研究的热点问题,也是 P2P 应用领域面临的核心问题之一.相关的非结构化 P2P 资源定位算法在查准率、查全率和查询成本上难以同时被优化,这会造成严重的网络带宽负担以及巨大的索引维护开销.为此,提出一种面向用户需求的非结构化 P2P 资源定位策略(user requirements resource location strategy,简称 U2RLS).该策略的创新点是:在原有非结构化 P2P 网络资源定位泛洪算法的基础上,融入用户需求、用户偏好、用户兴趣度等因素,首先进行用户资源子网划分;采用带有用户需求信息的泛洪和查询索引机制,对用户所需资源进行精确定位.该策略有效避免了因海量信息引起的网络风暴、信息重叠和资源搜索偏覆盖等问题,从而解决了查询节点盲目使用中继节点的现象.实验结果表明:面向用户需求的非结构化 P2P 资源定位策略 U2RLS 以其高搜索成功率、有限网络资源消耗和短查询时间响应等优势,能够显著地提高用户资源定位效率.

关键词: 用户需求;泛洪算法;资源定位;非结构化 P2P 网络;BFS&UPF

中图法分类号: TP393

中文引用格式: 何明,张玉洁,孟祥武.面向用户需求的非结构化 P2P 资源定位泛洪策略.软件学报,2015,26(3):640−662. http://www.jos.org.cn/1000-9825/4595.htm

英文引用格式: He M, Zhang YJ, Meng XW. Resource location flooding strategy of unstructured P2P for user requirements. Ruan Jian Xue Bao/Journal of Software, 2015,26(3):640−662 (in Chinese). http://www.jos.org.cn/1000-9825/4595.htm

Resource Location Flooding Strategy of Unstructured P2P for User Requirements

HE Ming[1,2], ZHANG Yu-Jie[1,2], MENG Xiang-Wu[1,2]

[1](Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing 100876, China)

[2](School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: In unstructured P2P networks, how to rapidly and precisely locate user required resources is currently a hot issue, and is also one of core problems faced by P2P application fields. Related unstructured P2P resources location algorithms can not be optimized at the same time in respect to precision ratio, recall ration and query costs, which can cause serious network bandwidth burden and huge index maintenance costs. To address the problem, this paper proposes a query strategy called user requirement resource location strategy (U2RLS). The innovation of this strategy is to integrate user requirements, user preferences and user interest based on the original unstructured P2P network resources location flooding algorithm. The strategy subnets the user resources, and adopts the flooding mechanism and query index mechanism with user required information to locate the resources accurately. This strategy effectively avoids the netstorm caused by mass information, data overlapping, and resource search partial coverage phenomenon, so as to solve the problem that query nodes use relay nodes blindly. The experimental results show that U2RLS has high search success rate, limited network

* 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: 2013-01-12; 修改时间: 2013-08-21; 定稿时间: 2014-03-27

resources consumption and short response time in query process, and therefore can significantly improve the efficiency for user resource location.

Key words: user requirement; flooding algorithm; resource location; unstructured P2P network; BFS&UPF

在非结构化的P2P网络中,能否有效地发现用户需求资源,已经成为目前影响网络整体性能的重要指标.P2P作为一种实现资源发现和共享的分布式技术,其核心思想是:系统依存于边缘化(非中央式服务器)设备的主动协助,每个用户直接从其他成员而不是服务器的参与中受益,用户在系统中同时扮演服务器与客户端的角色,但不能够意识到彼此的存在,用户之间在应用层建立虚拟连接,从而使整个系统中的所有用户节点互联组成了一个应用层上的逻辑虚拟网络[1].它充分利用系统中每一个用户节点来实现文件查询定位功能,使得处于网络边缘的用户更容易共享资源.而如何在用户节点之间高效、快速地定位用户所需资源,就成为近年来P2P领域研究的热点和核心问题.非结构化P2P网络对用户节点之间的关系比较随意,动态维护开销较小;对用户节点、资源和查询语义的表示更加多样化,它能够计算出查询用户节点及资源节点的相关性大小,从而使得查询能够以模糊匹配的方式进行.但是,非结构化P2P网络存在着用户需求资源定位效率低、准确性难以保证的问题[1].而泛洪搜索算法(flooding algorithm)[2]在搜索资源方面具有一定的代表性,它具有覆盖范围广、回应时间短、可靠性高等特点;该算法采用的是启发式广度优先搜索策略,一项研究表明[2],泛洪算法的搜索范围能够覆盖到 95%的用户节点.这是因为随着用户节点的增加,对查询信息进行路由转发,参与转发用户节点的个数呈指数级增长.另外,在采用泛洪式搜索算法的非结构化P2P系统中,用户节点的加入、离开以及失效等行为对整个系统所带来的负面影响非常小,因为节点担任的角色大致相同.这一点也说明了泛洪搜索方法的健壮性.然而,经典的泛洪定位算法不要求维护网络的拓扑结构,它是一种较直接的资源定位方法.

而在基于用户需求的环境下,泛洪算法极易出现网络风暴[2]和信息重叠[3]问题:

(1) 泛洪算法不考虑用户节点的状态,当进行资源定位时,节点均要转发数据包,致使节点因能量耗尽而死亡,影响网络的连通性;泛洪算法可以看成查询消息(query message)和查询命中消息(query hit message)在网络中被多次转发的过程[4],但当网络规模不断增大,每次路由都全网遍历,通过泛洪方式定位用户资源会造成网络流量急剧增加;

(2) 泛洪算法会出现一个用户节点可能多次接受同一个数据包的现象[5].因为泛洪模型中每一个用户节点,不论它是否在最终的定位路径上,都要转发数据包,直到含有该资源的节点接收到此数据包为止.这样就会导致在网络中充斥着大量无用的重复报文,使得节点的能量遭到了严重的损耗,整个网络的生存周期也受到影响.

针对泛洪算法的诸多缺陷,本文提出了一种面向用户需求的非结构化 P2P 资源定位策略 U2RLS.该策略的创新之处是:在原有非结构化 P2P 泛洪算法的基础上,依据用户需求信息对资源所在节点进行定位搜索.U2RLS 融入用户需求、用户偏好、用户兴趣度等因素:

- 首先,对用户资源所在节点进行子网相似度划分,结合用户需求度量,对泛洪路径作有针对性的选择.该度量并不是全网一致性的,而是基于用户需求的网络性能差分化度量;
- U2RLS 采用的是带有用户需求信息的泛洪策略和查询索引机制,其核心思想是:用户根据网络状态和实际需求对网络各项参数进行合理选择——利用带有用户需求的资源类型、用户偏好、资源存储位置等启发式索引信息,结合网络通信成本、网络连通性、查询返回定位结果数、用户共享资源数、用户资源查询跳数等网络参数信息,将资源定位消息向最有可能拥有目标资源的方向进行转发;
- 从图论与概率论的角度分析、证明了 U2RLS 可以有效地解决因海量信息引起的网络风暴、信息重叠和资源搜索偏覆盖等问题.改进了传统泛洪策略中 TTL 的计算方式,提出了 TTL 迭代深入策略,抑制了网络资源过度消耗的问题,解决了查询用户盲目使用中继节点等现象;
- 实验结果表明:与其他资源定位策略相比,U2RLS 具有用户资源定位成功率高、网络带宽消耗低和查询时间响应短等优势,在非结构化的 P2P 环境中,能够降低资源定位的广度与深度,显著地提高了用户资源定位效率,有效地避免了过多的路由消息转发而带来的网络拥堵现象.

1 相关工作

在资源定位过程中,客体对象是节点用户、节点资源和节点位置的统一体.节点用户是资源的拥有者,同时也是用户的需求者.而节点位置通过节点 ID 标识,总体呈现网络逻辑拓扑结构.为此,如何提高用户资源的定位效率,是非结构化 P2P 网络需要解决的首要问题.为了满足非结构化的 P2P 用户资源搜索定位的需要,研究者提出了许多解决方案,最典型的算法是泛洪算法和随机漫步(random walk)算法.另外,为了克服泛洪算法和随机漫步法的缺点,也提出了一些介于泛洪算法和随机漫步算法之间的用户资源定位方法.

文献[4]提出一种基于社会化网络特征的 P2P 资源定位策略,它利用社会化网络中的六度分隔基本原理和小世界模型实验推出结论:如果 P2P 网络的用户节点能够模拟六度分隔理论的查找方法,即可以最多通过 6 个用户节点查找到任意资源所在的节点,这就直接解决了查找深度问题.在小世界模型中,每个人传递信息时均发挥了自己的能动性,选择了自己认为更接近接收人的人来传递消息.通过模拟社会网络的特征,发挥节点的能动性,可以在有限的搜索深度和广度内快速查找定位文件.

文献[5]针对非结构化P2P用户资源定位算法Random Walk中过于依赖用户查询历史记录而导致动态网络环境下资源搜索命中率低、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,提出了双向随机漫步搜索机制.通过分析随机漫步的基本性质,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并形成双向搜索机制,在提高资源查询命中率的同时,不会产生过多的网络路由负担.并证明了其能够提高包括稀有资源在内的搜索成功率.

文献[6]对基于用户需求的泛洪算法进行了改进,主要在对邻居节点搜索时不再向所有邻居节点发送消息,而是根据相似性进行邻居节点选择.为了计算邻居节点的级别,查询接收节点将当前的查询和以前的查询进行比较,找出与以前查询的相似性,然后把查询继续发送到最可能的节点.文献[7]在无线传感器网络平台的基础上分析经典的泛洪路由算法,提出了改进型的概率递减式泛洪路由策略.

文献[8]提出了基于移动用户需求下局部观测概率论转播的泛洪算法,每个用户节点的转播概率都是时间和局部观测的一个函数.实验结果表明,该方法比其他方法在网络开销和搜索延迟上有明显的优势.文献[9]基于泛洪算法提出了一种有效自适应搜索机制(AFRA),该机制可以根据邻居转接节点的数量和 TTL(time to live)值的大小来满足不同用户的需求,提高了泛洪算法的灵活性.

此外,Gnutella[3]和KaZaA[10]用户资源定位方法也采用泛洪搜索机制,向所有的相邻节点发出查询请求,算法简单且搜索范围广,但网络开销比较大,同时需要限制搜索的深度.LinkNet[11]提出了一种可扩展分布式数据结构来支持大规模非结构化P2P系统的用户需求查找.PeerRank[12]提出基于概率权值的自适应查询消息转发策略,依据用户节点查询命中的历史信息,赋予节点相应权值作为查询消息路由选择的依据,引导查询消息快速接近目标资源.Gnutella2[13]提出了将节点分为超级节点和叶子节点的策略,它由少量超级节点和大量叶子节点组成,并在各个超级节点上存储了所辖叶子节点存储的文件索引,文件查找的主要任务就都在超级节点中完成,这样可以有效地减少转发查找请求的个数,降低网络负载.

2 资源定位泛洪模型及其数学语言描述

资源定位泛洪模型以传播速度较快、覆盖范围广、网络健壮性好、可靠性高而著称,其主要思想是:资源节点发布资源共享信息,该信息在网络中得到合理的传播与维护,模型采用启发式路由算法来定位用户所需资源.但资源定位泛洪模型产生大量的冗余消息,造成网络流量的急剧增加,并消耗大量带宽.由于用户对网络各项性能指标的偏好不同,进而引起搜索偏覆盖等问题.

2.1 非结构P2P网络资源定位泛洪模型

泛洪算法是一种典型的广度优先资源定位算法(broad flooding search,简称BFS)[14],其资源位置对于用户来说是未知的,当用户在非结构化P2P网络中搜索相关资源时,该用户节点向它的所有邻居节点广播资源查询消息,接到查询请求的邻居节点如果包含此用户所需资源,将资源命中信息返回用户节点;否则,邻居节点再向其

邻居节点泛洪资源定位请求.这个过程将持续进行下去.为了限制搜索范围,发起查询请求的用户在查询消息中设置一个初始的*TTL*值,查询消息向前广播 1 次,*TTL*值减少 1.如果*TTL*值减少至 0,资源仍没定位成功,则整个泛洪资源定位过程停止;如果资源定位成功,则向用户节点返回查询命中消息.在搜索过程中可能出现循环,但是由于*TTL*的控制,循环不会永远进行下去.其过程如图 1 所示.

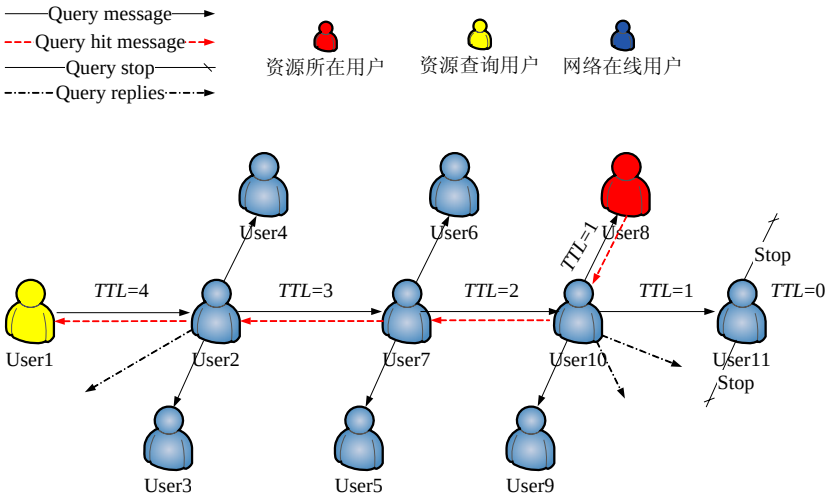


Fig.1 User resources location map

图 1 用户资源定位图

2.2 泛洪算法模型描述

2.2.1 泛洪算法模型数学语言描述

在Newman网络拓扑结构的环境中,采用Power law随机图[15]和Bimodal[16]拓扑图来评价泛洪资源定位算法的各项性能指标.

首先,用$G_0(x)$表示k度节点分布函数:

$$G_0(x)=\sum_{k=1}^{m}P_k x^k \tag{1}$$

其中,P_k是随机选取k度节点的概率,m是节点的最大度数.

基于这个生成函数,随机选取节点的平均度数为 $z_1=\sum_{k=1}^{m}kp_k=G_0'(1)$.这样,下一个邻居节点的数量为

$$z_2=\left[\frac{\mathrm{d}}{\mathrm{d}x}G_0(G_1(x))\right]_{x=1}=G_0'(1)G_1'(1).$$

这里,$G_0'(1)$ 是$G_0(1)$的一阶导数,而 $G_1(x)=\dfrac{G_0'(x)}{G_0'(1)}$,$G_1'(1)$ 是$G_1(1)$的一阶导数.

网络模型 1(Power law随机图). Power law随机图节点度数服从τ的指数分布,公式(1)中的P_k与$k^{-\tau}$成正比,$G_0'(1)$ 的值近似值为 $(1-m^{2-\tau})/(\tau-2)$,$G_1'(1)$ 的近似值为 $m^{3-\tau}/(G_0^1(1)\cdot(3-\tau))$,$\tau\in(2,3)$, m 为节点数.

网络模型 2(Bimodal拓扑图). 在Bimodal拓扑图中,将用户节点分成超级节点和普通邻居节点,少量超级节点$T_1=\{A_1,A_2,A_3,\ldots,A_n\}$和大量邻居节点相邻,而其余超级节点$T_2=\{B_1,B_2,B_3,\ldots,B_m\}$与少量邻居节点相邻.随机选取节点时,选取$T_1$中节点的概率为$P_{ultra}$,选取$T_2$中节点的概率为$P_{few}$,$P_{few}=1-P_{ultra}$.$T_1$中节点的度数表示为$K_{ultra}$, T_2中节点度数表示为K_{few},将P_{ultra},P_{few},K_{ultra},K_{few}利用公式(1)中的z_1,z_2得出每条所查询的平均节点数量.

定理 1(资源定位成功率$SR_{flooding}$). 如果用户所需资源是以复制比R分布于网络中,那么资源定位成功率为公式(2);资源定位成功率SR与网络覆盖率成正比:

$$SR_{flooding}=1-(1-R)^C \tag{2}$$

证明:R为同一资源在网络各用户节点的分布率,C为网络覆盖率,$R\in(0,1)$.随着覆盖率C的增加,$(1-R)^C$的值减小,则$SR_{flooding}$的值逐渐增大,即资源定位成功率SR与网络覆盖率成正比. □

定理 2(资源定位搜索时间$ST_{flooding}$). 如果成功查询用户资源所用时间为$ST_{flooding}$,那么它由用户资源成功查询所需最少时间SR_{req}和网络覆盖率C共同决定:

$$ST_{flooding}=\log_{p\cdot G_1'(1)}\left(\frac{(p\cdot G_1'(1)-1)\cdot\log_{(1-R)}(1-SR_{req})}{p\cdot G_0'(1)}+1\right) \tag{3}$$

证明:从 $G_0'(1),G_1(1)$ 公式中,得到有限变换集合 $\{G_0'(1),G_1'(1),(G_1'(1))^2,...,(G_1'(1))^{ST_{flooding}-1}\}$.

根据 $G_0'(1)\cdot(G_1'(1))^{t-1}\leqslant k\leqslant G_0'(1)\cdot(G_1'(1))^t$,当网络覆盖率$C=\log_{(1-R)}(1-SR_{req})$时,公式(4)两端同时取对数:

$$p\cdot G_0'(1)+p^2G_0'(1)\cdot G_1'(1)+p^3\cdot G_0'(1)\cdot(G_1'(1))^2+...+p^{ST_{flooding}}\cdot G_0'(1)\cdot(G_0'(1))^{ST_{flooding}-1}=\log_{(1-R)}(1-SR_{req}) \tag{4}$$

得出公式(3)$ST_{flooding}$. □

定理 3(资源查询命中率$QH_{flooding}$). 如果用户所需资源是以复制比R均匀分布于网络中,网络覆盖率为C,用V_h表示一次资源查询定位事件,在第h跳访问一个节点,则查询命中率为$QH_{flooding}=R\cdot C$.由此可得公式(5):

$$C_h=\begin{cases}\sum_{i=1}^{n}P_i(V_h), & h=1\\ \sum_{i=1}^{n}\prod_{j=1}^{h-1}[1-P_i(V_j)]\cdot P_i(V_h), & h\geqslant 2\end{cases} \tag{5}$$

证明:查询命中率是由网络覆盖率C决定,节点I在第h跳时被访问的概率为$P_i(V_h)$,用C_h表示第h跳时的网络覆盖率,当跳数h为 1 时,C_h为第 1 跳访问节点的期望;当跳数h大于 1 时,C_h是当前整个网络覆盖率.即有公式(6)成立:

$$P_i(V_h)=\begin{cases}p\cdot p_i\cdot G_0'(1), & h=1\\ 1-[1-p\cdot p_i\cdot G_1'(1)]^{C_{h-1}}, & h\geqslant 2\end{cases} \tag{6}$$

这里,P_i为访问节点i的概率.

文献[17]指出, $P_i=(m/i^{1\tau})/\left(\sum_{i=1}^{n}m/i^{1\tau}\right)$,$\tau$是Power law随机图的分布指数,$m$是节点最大度数.$P_i(V_h)$是由$h$和$P_i$共同决定,因此可得公式(5). □

定理 4(资源查询消息数$QM_{flooding}$). 如果资源查询路由跳数为h,第h跳的覆盖率为C_h,那么在第h跳时的查询消息(query message)数量e_h为公式(7):

$$e_h=\begin{cases}p\cdot G_0'(1), & h=1\\ p\cdot G_1'(1)\cdot C_{h-1}, & h\geqslant 2\end{cases} \tag{7}$$

证明:当查询消息数的每一跳赋值为 k 时,总查询跳数为 $TTL\times k$,此时,广度优先搜索算法的查询消息数为公式(8):

$$e_h=\begin{cases}p\cdot G_0'(1), & h=1\\ p\cdot G_1'(1)\cdot C_{h-1}, & 2\leqslant h\leqslant n\\ C_n, & h>n\end{cases} \tag{8}$$

而泛洪算法的跳数并不受限于覆盖率C_n,因此将C_n忽略不计,得到公式(7). □

定理 5(资源定位查询效率$QE_{flooding}$). 如果h跳的查询命中率为$QH(h)$,QM是查询过程中所有消息总数,R为资源在网络中的复制比,那么第h跳的资源定位查询效率$QE_{flooding}$为公式(9):

$$QE_{flooding}=\frac{\sum_{h=1}^{TTL}QH(h)/h}{QM}\cdot\frac{1}{R} \tag{9}$$

证明:资源定位查询的终极目标并不是覆盖到全网节点,而是搜索到目标节点和资源.网络覆盖率C_h是充分条件,因此定义$QH(h)$为第h跳的查询命中率.在定理 3 中证明过,它是通过访问概率$P_i(V_h)$和跳数h定义搜索

范围和搜索深度,用 $\sum QueryHit(h)/QueryMsg$ 定义查询效率,此公式计算出来的结果误差较为严重,因此引入复制比 R 进行补偿,总资源定位查询效率为

$$QE'_{flooding}=\left(\sum_{h=1}^{TTL}QH(h)\Big/QM\right)\cdot\frac{1}{R}.$$

则第h跳的$QE_{flooding}$为公式(9). □

定理 6(资源定位搜索效率$SE_{flooding}$). 如果已知深度为h的节点资源查询命中率为$QH(h)$,资源定位成功率为SR,资源复制比为R,资源查询消息数为QM,那么资源定位搜索效率为公式(10):

$$SE_{flooding}=\frac{\sum_{h=1}^{TTL}QH(h)}{QM\cdot h}\cdot\frac{SR}{R} \tag{10}$$

证明:泛洪算法在二叉树中进行资源定位搜索不产生任何冗余信息,则资源定位搜索效率为公式(11).由此:

$$SE_{flooding}=\frac{\sum_{h=1}^{TTL}E(x)_h/h}{k\times TTL}\times\left[1-(1-R)^{\sum_{h=1}^{TTL}E(x)_h}\right] \tag{11}$$

推广到Power law随机图中,每个节点被访问的概率为$P_i=1/2^h$.这样,第k跳访问一个节点的逆概率为$(1-1/2^h)^k$,深度为h,在第k跳访问的平均节点数学期望为$E_h(x)=2^h[1-(1-1/2^h)^k]$,而$QH(h)=R\cdot E_h(x)$,$QM=k\cdot TTL$,则有:

$$QE_{flooding}=\left(\sum_{t=1}^{TTL}R\cdot E(x)\right)/(K\cdot TTL\cdot R)=\left(\sum_{t=1}^{TTL}E(x)t\right)/QM\ .$$

而由定理 1 知, $SR=1-(1-R)^C=1-(1-R)^{\sum_{h=1}^{TTL}E_h(x)}$,带入公式(11)中,公式(10)得证. □

2.2.2 盲目泛洪算法问题描述

问题 1(网络风暴). 泛洪模型不考虑节点能量可用状况,无论此时节点的能量如何,当进行泛洪广播时,节点均要转发数据包,这会导致节点因能量耗尽而死亡.对于更大规模的网络,则随着消息跳数的增加,网络节点数的增多,一次泛洪查询所带来的总的开销会更大;而且当网络中节点数增加时,节点发起的查询请求数目也同时增加,网络的整体流量迅速增大,从而引起网络风暴.

问题 2(信息重叠). 出现一个节点可能多次接收同一个数据包的现象.因为泛洪模型中的每一个节点,不论它是否在最终的转发路径上都要转发数据包,直到目的节点接收到此数据包为止,这就会导致在网络中充斥了大量无用的重复报文,使得节点的能量遭到了严重的损耗,整个网络的生存周期也倍受影响.

问题 3(搜索偏覆盖). 文献[18]表明:采用某种启发式规则的泛洪策略在进行资源搜索时会产生搜索偏覆盖问题,有些资源节点相对用户查询节点来说是覆盖范围之外的盲点.当用户在资源定位时,采用这种策略无法搜索到这些节点上的资源.

如图 2(a)所示,假设用户选择路由标准是资源数量,用户A在泛洪定位时,向它的邻居节点集$T_A=\{B,C,D,E\}$中的节点发送查询消息,C的资源数量最大,A向C发送,C向它的邻居节点集$T_C=\{B,D,G\}$中的节点发送查询消息,G含有的资源数最大,C向G发送,G向它的邻居节点集$T_G=\{F,M\}$中的节点发送查询消息,F含有的资源数最大,G向F发送,F向它的邻居节点集$T_F=\{B,H\}$中的节点发送查询消息,B含有的资源数最大,F向B发送,B向它的邻居节点集$T_B=\{A,C\}$中的节点发送查询消息,A含有的资源数最大,B向A发送,由此可得查询路径为$A\to C\to G\to F\to B\to A$.在整个查询过程中,只有$A,C,G,F,B$这 5 个节点被覆盖,其余节点对于用户$A$是不可见的.

如图 2(b)所示,假设用户选择路由标准是网络连通性,用户A在泛洪定位时向它的邻居节点集$T_A=\{B,C,D,E\}$发送查询消息,路径AD的连通性最优,A向D发送,D向它的邻居节点集$T_D=\{C,E,I,J,K\}$发送查询消息,路径DK的连通性最优,D向K发送,K向它的邻居节点集$T_K=\{L\}$发送查询消息,由于只有一条路径即KL,K向L发送,由此可得查询路径为$A\to D\to K\to L$.在整个查询过程中,只有A,D,K,L这 4 个节点被覆盖,其余节点对于用户A仍然是不可见的.

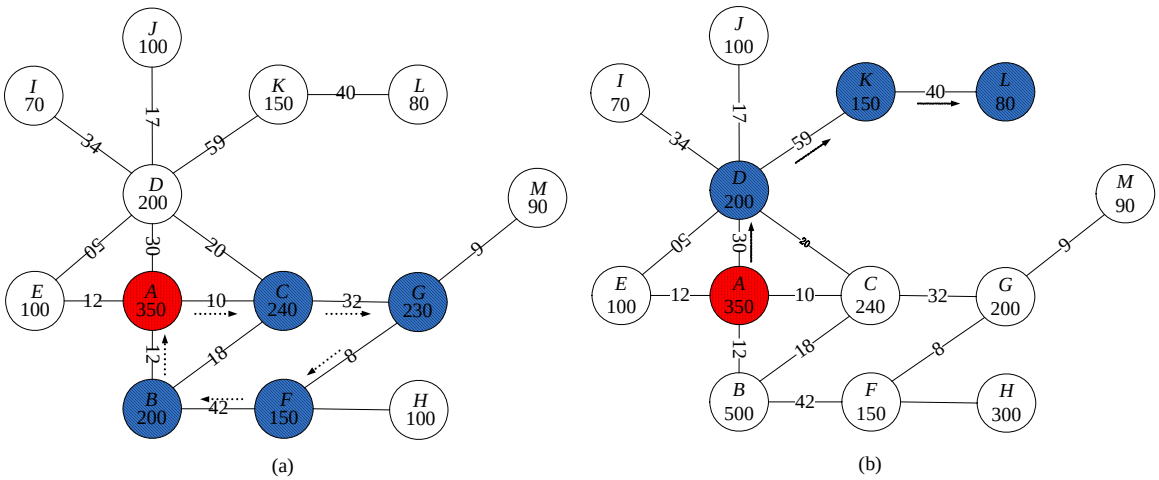


Fig.2 Partial coverage problem of flooding

图 2 Flooding 算法搜索偏覆盖问题

3 面向用户需求的资源定位 U2RLS 泛洪策略

U2RLS 是一种基于用户需求度量的资源定位策略,该度量并非是全网一致性的,而是基于用户需求的网络性能差分化度量.其核心思想是:用户根据网络状态和实际需求对网络各项参数进行合理选择,U2RLS 利用带有网络连通性、用户共享资源数、用户返回资源定位结果等用户需求启发式索引信息,将资源定位消息向最有可能拥有的目标节点方向进行转发,有效地避免过多的路由转发消息带来的网络拥堵现象.

3.1 用户兴趣本体

利用兴趣概率树可以有效地建立启发式索引信息,而该类索引信息同步速度较快,针对完全分散式的非结构化 P2P 网络拓扑结构,可以有效防止网络风暴和信息重叠现象的产生.

定义 1(本体). 本体是指对存在本质的研究[19].本体是描述概念及概念之间关系的模型,通过概念分析、建模,把现实世界中的事物表示成为一种有效的概念层次结构和语义模型.

定义 2(本体树). 每个节点表示一个独立的概念,每一条边表示从父节点指向子节点的有向边,子节点是父节点的子概念[19].如图 3 所示.

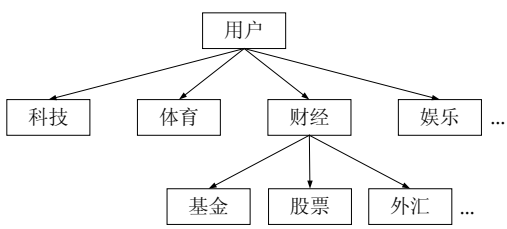


Fig.3 Ontology tree

图 3 本体树

定义 3(兴趣概率). 用户节点X对用户节点Y所拥有的资源兴趣程度[19].

定义 4(兴趣概率树). 兴趣概率树表示用户兴趣的概率树形分布,是用来揭示具有层次结构的本体随机输出的数学结构[19].如图 4 所示,假设用户A对用户B和C的兴趣概率分别为 0.8 和 0.2,用户B对用户D和E的兴趣概率为 0.9 和 0.1,用户C对用户F和D的概率各为 0.5.由此可以得到如下定理:

定理 7(兴趣偏差概率). 如果概率树中存在从节点 X 到节点 Y 之间的有向通路,那么 X 对 Y 的兴趣概率是

这两点之间所有有向边概率的乘积,如图 4 中 $P(E|A)=P(B|A)P(E|B)=0.1\times0.8=0.08$;如果概率树中不存在从节点 X 到 Y 之间的有向通路,则从 X 到 Y 之间的兴趣概率为 $P(Y|X)=0$,如图 4 中 $P(G|B)=0$.

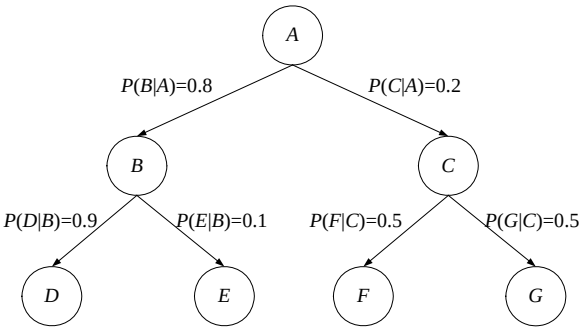


Fig.4 Probability tree

图 4 概率树

定义 5(兴趣偏差方程). 兴趣偏差方程用来衡量两个兴趣本体概率树分布之间的偏差程度.若 P 和 Q 表示两个兴趣本体,则它们的概率树分布之间的兴趣偏差方程为公式(12):

$$D(P\|Q)=\begin{cases}\sum_{i=1}^{N}P_{x_i}\log\dfrac{P_{x_i}}{Q_{x_i}}, & Q_{x_i}\neq 0\\ \infty, & Q_{x_i}=0\end{cases} \tag{12}$$

其中,N 为 P 和 Q 两个兴趣本体概率树有向边的并集,P_{x_i} 和 Q_{x_i} 分别是本体 P 和本体 Q 兴趣概率树对应的有向边的概率值.

3.2 用户兴趣子网划分算法

假设系统中所有节点都使用相同类型的用户兴趣本体Q_s.资源在每个节点上是以资源路由表建立索引文件,例如大小、数量、类型、存储路径、兴趣概率值等资源信息.用户S开始进行资源定位时,并没有系统中其他用户兴趣概率信息,采用基于用户需求的BFS定位算法,查询消息到达资源节点A后,首先查询该节点的资源路由索引表,如果该节点资源本体与用户S本体的兴趣偏差值小于阈值r(兴趣偏差阈值),查询命中消息将节点A上的资源索引信息返回用户S,并根据兴趣概率值生成用户S的兴趣概率树,用户兴趣概率树是由每一个返回该用户的查询命中消息所携带的兴趣概率值构建起来的.BFS算法结束时,用户S将收到拥有该资源相关信息的一组节点$PG(S)=\{A_1,A_2,A_3,\ldots,A_n\}$发出的查询命中消息,对该消息中的兴趣概率值进行提取,使其能够构建出资源节点的兴趣概率树.

算法 1. 用户兴趣子网划分算法.

Input: the set of resource nodes: $PG(S)=\{A_1,A_2,A_3,\ldots,A_n\}$, interest diversity threshold value: r=0.3, initial value of deviation distance: μ, the probability of X is interested in Y: $P(Y|X)$, cycle control number of nodes: i=0, cycle control number of times: j=0, interest subnet nodes set of S:$ISN(S)=\varnothing$;

Output: $ISN(S)=\{A_1',A_2',A_3',...,A_n'\}$.

1. **for** each cycle j of the times { //Periodically structure of user interest subnet.
2. **for** each resource nodes A_i { //Calculate the deviation distance between user node S and resource node A_i in $PG(S)$.
3. $\mu=D(S\|A_i)$; //A_i is belong to $PG(S)$ and $P(Y|X)$ is belong to interest Probability tree
4. **if** ($\mu<r$) then ($A_i\in ISN(S)$);
5. **else if** (A_i is already in $ISN(S)$) **then** delete A_i from $ISN(S)$;
6. **else** delete A_i from $PG(S)$;}

7. $ISN(S)=\{A'_1,A'_2,A'_3,...,A'_n\}$ //Reconstruct subnet according to $ISN(S)$
8. UPF algorithm will run in $ISN(S)=\{A'_1,A'_2,A'_3,...,A'_n\}$; //UPF algorithm will be introduced in next section.}

由于基于用户需求的 BFS 定位搜索运行在算法 1 之前,相对于纯粹的 BFS 搜索,前者所产生查询消息数较小,网络开销也相对较低.兴趣概率树是围绕着基于用户需求的 BFS 算法搜索过程建立的,与此同时,用户 S 兴趣资源节点集 $ISN(S)=\{A'_1,A'_2,A'_3,...,A'_n\}$ 对应的兴趣概率集 $P(S)=\{P_1,P_2,P_3,\ldots,P_n\}$ 生成兴趣概率树.而资源索引算法的时间复杂度相对较低,占用的空间极为有限.故,节点维护与其他所有节点的兴趣概率树的开销是在可接受的范围之内.随着资源节点数在一定范围内增加,查询效率随之明显增大;当节点数超过一定范围时,算法的查询效率下降.这是因为网络节点数不断增多,维护资源索引文件一致性开销将急剧增大,影响到整个系统的定位性能.当一个资源节点加入网络时,系统需要考虑维护该资源索引文件的开销,即用户兴趣概率树的一致性维护开销,它包含 3 个部分[20]:(1) 一个节点资源更新后,更新原文件的开销;(2) 原文件更新后,在含有该资源信息的所有节点上,对其兴趣概率值一致性的开销;(3) 资源索引文件的更新,以便建立新的兴趣概率树的开销.

3.3 混合度量周期性泛洪策略

定义 6(周期性泛洪函数). 基于概率统计的一种资源定位策略,在泛洪过程中,中继节点的数量 h 随着泛洪路径的扩展而改变,而 h 值是由 TTL 函数来控制,即:

$$h=f(n,TTL).$$

由上式可知,h是随着邻居节点数n和TTL值变化的周期函数[18]:

$$h=f(n,TTL)=\begin{cases}\left\lceil\frac{1}{2}n\right\rceil, & TTL \text{ is odd}\\ \left\lceil\frac{1}{3}n\right\rceil, & TTL \text{ is even}\end{cases} \tag{13}$$

在BFS算法和深度优先搜索算法DFS(deepen first search)中,中继节点h的数量是不变的,分别为$h=f_{BFS}(n,TTL)=n$和$h=f_{DFS}(n,TTL)=1$.在整个查询定位过程中,周期性泛洪机制是分成若干个阶段,周期性地循环,将不同循环阶段的数量定义为C.

定义 7(用户需求度量metrics). 在利用U2RLS策略进行资源定位时,本策略根据用户需求来进行资源定位,其包括[18]网络通信成本、网络连通性、查询返回定位结果数量、用户共享资源数量、用户资源查询跳数等,用户需求根据不同种类被分配相应的权重值w_i,为了获得优异的资源定位结果,用户可以将历次查询效果较好的度量赋予较高的权重.这里,$\sum_{i=1}^{t}w_i=1$,w_i是度量i的权重,且 $1\leqslant i\leqslant t$.

定义 8(用户需求度量周期泛洪函数). 为了减小偏覆盖问题,引入第 i 种度量下混合度量周期泛洪函数:

$$h_i=\lceil h\times w_i\rceil.$$

这里,w_i为用户选取的第i种度量[18].

算法 2. 用户需求度量周期泛洪算法.

Input: the number of neighbors: n, the number of A_i's flooding nodes: h_{A_i},h'_{A_i}, the set of TTL value: $TTL=\{a,b,c,\ldots,m,n\}$, user node: S, the set of mix metrics in the network: $W(S)=\{w_1,w_2,w_3,\ldots,w_m\}$, the set of mix metrics chosen by user S: $W'(S)=\{w'_1,w'_2,w'_3,...,w'_m\}$, the set of S's interest subnet nodes: $ISN(S)=\{A'_1,A'_2,A'_3,...,A'_n\}$, empty set 1: $H=\Phi$. empty set 2: $H'=\Phi$;

Output: the number of queries: m, the number of nodes which receive new messages: u, the probability of nodes which receive messages: p.

1. **for** each TTL a {
2. **for** each neighbor A_i { //Calculate the number of A_i's flooding nodes h_{A_i}
3. **if** (a is odd), **then** ($h_{A_i}=\lceil n/2\rceil$);

4. **else** ($h_{A_i}=\lceil n/3\rceil$);
5. $h_{A_i}\in H\{h_{A_1},h_{A_2},h_{A_3},...,h_{A_n}\}$;
6. **for** each h_{A_i} of A_i { //Calculate the total number h'' of flooding nodes according to user metric.
7. **for** each metric w_i of S { //Calculate the flooding nodes number h_{A_i} according every user metric w_i in $W(S)$.
8. $h'_{A_i}=\lceil h_{A_i}\times w_i\rceil$;
9. $h'_{A_i}\in H'$;}
10. $h''=h'_{A_1}+h'_{A_2}+h'_{A_3}+...+h'_{A_m}$;
11. **for** each metric w_i of S { //Choose the nodes user required to send query messages according to each user metric.
12. **for** each neighbor A_i { //Choose the nodes user required to send query messages according to each flooding node.
13. **if** (w'_i of A_i in $H'>w_i$) **then** (send the Query message to $A_i\in ISN(S)$); //if neighbor $A_i\notin ISN(S)$, query messages will not be sent to it.
14. **else** (delete A_i from H);}
15. $H=\Phi$; $H'=\Phi$ //Clear the set of H.
16. if (TTL=0) then (end the frist for loop); //Use TTL to control loop times.
17. else ($TTL=TTL-1$);}}

3.3.1 BFS 算法和 UPF 算法验证网络风暴、信息重叠、网络偏覆盖过程的对比分析

为了说明用户需求周期性算法相对于 BFS 算法可以避免网络风暴、盲目使用资源以及搜索偏覆盖等问题,利用图论和概率论等思想加以证明.

证明:(BFS 算法的泛洪路由过程及消息数量).

(1) $TTL=7, RN(S)=\{A,B,C,D\}$. 消息发送路径为$S\to A, S\to B, S\to C, S\to D$.

(2) $TTL=6, RN(A)=\{B,D,I,Q\}, RN(B)=\{C,L,K,W,J\}, RN(C)=\{F,G,L,B\}, RN(D)=\{A,E,F\}$.

消息发送路径为$S\to A\to B, S\to A\to D, S\to A\to I, S\to A\to Q$;

$S\to B\to C, S\to B\to L, S\to B\to K, S\to B\to K, S\to B\to J, S\to B\to X$;

$S\to C\to F, S\to C\to G, S\to C\to L, S\to C\to B$;

$S\to D\to A, S\to D\to E, S\to D\to F$.

(3) $TTL=5, RN(B)=\{J,W,K,L,C,S\}, RN(D)=\{S,E,F\}, RN(I)=\{J,H,Q\}, RN(Q)=\{R,E,I\}$;

$RN(C)=\{S,F,G,L\}, RN(L)=\{M,C\}, RN(K)=\{X\}, RN(W)=\phi, RN(J)=\{I,O\}$;

$RN(F)=\{G,D,T\}, RN(G)=\{F,Y\}, RN(L)=\{B,M\}, RN(B)=\{A,S,J,W,K,L\}$;

$RN(A)=\{Q,S,I,B\}, RN(E)=\{Q,Z\}, RN(F)=\{C,G,T\}$;消息发送路径为

$S\to A\to B\to J, S\to A\to B\to W, S\to A\to B\to K, S\to A\to B\to L, S\to A\to B\to C,$

$S\to A\to B\to S; S\to A\to D\to S, S\to A\to D\to E, S\to A\to D\to F; S\to A\to I\to J,$

$S\to A\to I\to H, S\to A\to I\to Q; S\to A\to Q\to R, S\to A\to Q\to Z, S\to A\to Q\to I;$

$S\to B\to C\to S, S\to B\to C\to F, S\to B\to C\to G, S\to B\to C\to L, S\to B\to L\to M,$

$S\to B\to L\to C, S\to B\to K\to X, S\to B\to J\to I; S\to B\to J\to O; S\to C\to F\to G,$

$S\to C\to F\to D, S\to C\to F\to T, S\to C\to G\to F, S\to C\to G\to Y, S\to C\to L\to B,$

$S\to C\to L\to M, S\to C\to B\to S, S\to C\to B\to A, S\to C\to B\to J, S\to C\to B\to W,$

$S\to C\to B\to K, S\to C\to B\to L; S\to D\to A\to Q, S\to D\to A\to S, S\to D\to A\to I,$

$S\to D\to A\to B, S\to D\to E\to Z, S\to D\to F\to C, S\to D\to F\to G, S\to D\to F\to T.$

(4) $TTL=4, RN_1(J)=\{I,O\}, RN_1(W)=\phi, RN_1(K)=\{X\}, RN_1(L)=\{C,M\}, RN_1(S)=\{A,C,D\};$

$RN_2(S)=\{A,B,C\}, RN_1(E)=\{Q,E\}, RN_1(F)=\{C,T,G\}; RN_2(J)=\{B,O\}, RN_1(H)=\{N,P\},$

$RN_1(Q)=\{A,E,R\}; RN_1(R)=\phi, RN_2(E)=\{Z,D\}, RN_1(I)=\{A,H,I\}; RN_3(S)=\{A,B,D\},$

$RN_2(F)=\{D,G,T\}, RN_1(G)=\{F,Y\}, RN_2(L)=\{B,M\}; RN_1(M)=\{U\}, RN_1(C)=\{B,S,F,G\};$

$RN(X)=\phi; RN_2(I)=\{A,H,Q\}, RN(O)=\{V\}; RN_2(G)=\{C,Y\}, RN(D)=\{A,S,E\}, RN_1(T)=\phi;$

$RN_3(F)=\{C,D,T\}, RN(Y)=\phi; RN_1(B)=\{K,W,J,A,S,C\}, RN_2(M)=\{U\}; RN_4(S)=\{A,C,D\},$

$RN(A)=\{I,Q,D,S\}, RN_3(J)=\{I,O\}, RN_2(W)=\phi, RN_2(K)=\{X\}, RN_3(L)=\{M\};$

$RN_2(Q)=\{I,R,E\}, RN_4(S)=\{B,C,D\}, RN_4(I)=\{H,Q,J\}, RN_2(B)=\{J,W,K,L,C,S\};$

$RN_3(Q)=\{R,A,I\}, RN(Z)=\phi; RN_2(C)=\{S,B,G,L\}, RN_2(G)=\{C,Y\}, RN_2(T)=\phi;$消息发送路径为

$S\to A\to B\to J\to I,O; S\to A\to B\to K\to X; S\to A\to B\to L\to C,M;$

$S\to A\to B\to S\to A,C,D; S\to A\to D\to S\to A,B,C; S\to A\to D\to E\to Q,Z;$

$S\to A\to D\to F\to C,T,G; S\to A\to I\to J\to B,O; S\to A\to I\to H\to N,P;$

$S\to A\to I\to Q\to A,E,R; S\to A\to Q\to E\to Z,D; S\to A\to Q\to I\to H,A,J;$

$S\to B\to C\to S\to A,B,D; S\to B\to C\to F\to D,T,G; S\to B\to C\to G\to F,Y;$

$S\to B\to C\to L\to B,M; S\to B\to L\to M\to U; S\to B\to L\to C\to B,S,F,G;$

$S\to B\to J\to I\to H,Q,A; S\to B\to J\to O\to V; S\to C\to F\to G\to C,Y;$

$S\to C\to F\to D\to A,S,E; S\to C\to G\to F\to C,D,T; S\to C\to L\to B\to K,W,J,A,S,C;$

$S\to C\to L\to M\to U; S\to C\to B\to S\to A,C,D; S\to C\to B\to A\to I,Q,D,S;$

$S\to C\to B\to A\to I,Q,D,S; S\to C\to B\to J\to O,I; S\to C\to B\to K\to X;$

$S\to C\to B\to L\to M; S\to D\to A\to Q\to I,R,E; S\to D\to A\to S\to B,C,D;$

$S\to D\to A\to I\to H,Q,J; S\to D\to A\to B\to J,W,K,L,C,S; S\to D\to E\to Q\to A,R,I;$

$S\to D\to F\to C\to S,G,B,L; S\to D\to F\to G\to C,Y;$

BFS 算法路由过程如图 5 所示.

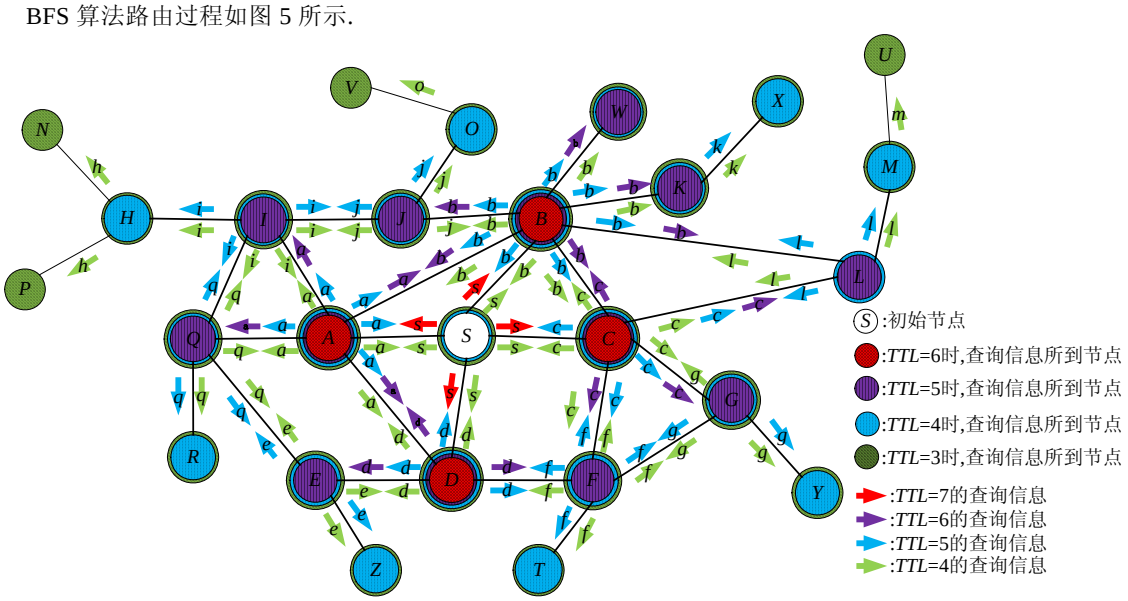


Fig.5 BFS algorithm message routing plan

图 5 BFS 算法消息转发图

假设用户节点 S 发起查询请求,查询消息要经过 3 次泛洪到达所有节点,整个过程产生 110 条查询消息.其

中,有 4 条路径查询消息只经过一次;而其余 30 条路径,同一查询消息多次经过,并产生了 76 条冗余消息,当每个路径最多限制一定数量查询消息经过时,网络风暴产生.

在 UPF 算法中,用户节点 S 发起查询请求:

$$ISN(S)=\{A_1',A_2',A_3',...,A_n'\},TTL=7.$$

用户需求度量:网络连通性w_1=0.5,节点共享资源数量w_2=0.5.

在如下的证明中,同时描述了 $ISN(S)$的构建过程.

证明(UPF 算法的泛洪路由过程及消息数量):

(1) $TTL=7,RN(S)=PG(S)=\{A,B,C,D\},n_s=4,w_1=0.5,w_2=0.5.$

$h^S=\lceil 4/2\rceil=2,h_1^S=1,h_2^S=1,h^{S'}=h_1^S+h_2^S=2$,选择$A,C$. $ISN(S)=\{S,A,C\}$.

(2) $TTL=6,RN(A)=\{B,I,Q,D\},RN(C)=\{B,F,L,G,F\},n_A=4,n_C=4.$

$h^A=\lceil 4/3\rceil=2,h_1^A=1,h_2^A=1,h^{A'}=h_1^A+h_2^A=2$,选择$I,Q.;h^C=\lceil 4/3\rceil=2,h_1^C=1,h_2^C=1,h^{C'}=h_1^C+h_2^C=2,$

选择$B,G.ISN(S)=\{S,A,B,C,G,I,Q\}$.

(3) $TTL=5,RN(B)=\{K,L,J,W\},RN(G)=\{F,Y\},RN(I)=\{H,J\},RN(Q)=\{R,E\},n_B=6,n_G=2,n_I=3,n_Q=3.$

$h^B=\lceil 6/2\rceil=3,h_1^B=2,h_2^B=2,h^{B'}=h_1^B+h_2^B=4$,选择$K,L,J,W$;

$h^G=\lceil 2/2\rceil=1,h_1^G=1,h_2^G=1,h^{G'}=h_1^G+h_2^G=2$,选择$F,Y$.

$h^I=\lceil 3/2\rceil=2,h_1^I=1,h_2^I=1,h^{I'}=h_1^I+h_2^I=2$,选择$H,J$;

$h^Q=\lceil 3/2\rceil=2,h_1^Q=1,h_2^Q=1,h^{Q'}=h_1^Q+h_2^Q=2$,选择$R,E$.

$ISN(S)=\{S,A,B,C,E,F,G,H,I,J,K,L,Q,R,W,Y\}$.

(4) $TTL=4,RN(R)=\phi,RN(E)=\{D,Z\},RN(H)=\{P,N\},RN_1(J)=\{O\},RN_2(J)=\{O\}$(注:节点$J$,在$TTL=4$时,同时接受节点$I,O$的消息),$RN(W)=\phi,RN(L)=\{M\},RN(K)=\{X\},RN(F)=\{D,T\},RN(Y)=\phi,n_R=0,$

$n_E=2,n_H=2,n_{J_1}=2,n_{J_2}=2,n_W=0,n_L=2,n_k=1,n_F=3,n_Y=0.$

$h^E=\lceil 2/3\rceil=1,h_1^E=1,h_2^E=1,h^{E'}=h_1^E+h_2^E=2$,选择$Z,D$;

$h^F=\lceil 3/3\rceil=1,h_1^F=1,h_2^F=1,h^{F'}=h_1^F+h_2^F=2$,选择$D,T$.

$h^H=\lceil 2/3\rceil=1,h_1^H=1,h_2^H=1,h^{H'}=h_1^H+h_2^H=2$,选择$P,N$;

$h^{J_1}=\lceil 2/3\rceil=1,h_1^{J_1}=1,h_2^{J_1}=1,h^{J_1'}=h_1^{J_1}+h_2^{J_1}=2$,选择 O^B.

$h^{J_2}=\lceil 2/3\rceil=1,h_1^{J_2}=1,h_2^{J_2}=1,h^{J_2'}=h_1^{J_2}+h_2^{J_2}=2$, 选择$O^I$;

$h^K=\lceil 1/3\rceil=1,h_1^K=1,h_2^K=1,h^{K'}=h_1^K+h_2^K=2$, 选择$X$.

$h^L=\lceil 2/3\rceil=1,h_1^L=1,h_2^L=1,h^{L'}=h_1^L+h_2^L=2$,选择$M$;

$ISN(S)=\{S,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,T,W,X,Y,Z\}$.

(5) $TTL=3,RN_1(D)=\phi,RN_2(D)=\phi$(注:节点$D$,在$TTL=4$时,同时接受节点$E,F$的消息),

$RN(B)=\phi,RN(O)=\{V\},RN(I)=\phi,RN(M)=\{U\},RN(C)=\phi,$

$RN(P)=\phi,RN(N)=\phi,RN(Z)=\phi,RN(N)=\phi,$

$RN(X)=\phi,n_{D_1}=3,n_{D_2}=3,n_B=6,n_O=1,n_I=3,n_M=1,n_C=4,n_P=0,n_N=0,n_Z=0,n_X=0,n_Z=0,n_P=0.$

$h^O=\lceil 1/2\rceil=1,h_1^O=1,h_2^O=1,h^{O'}=h_1^O+h_2^O=2$,选择$V$(注:只有一个节点可以选择).

$h^M=\lceil 1/2\rceil=1,h_1^M=1,h_2^M=1,h^{M'}=h_1^M+h_2^M=2$,选择$U$(注:只有一个节点可以选择).

$ISN(S)=\{S,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,T,U,V,W,X,Y,Z\}$.

如图 6 所示,UPF 算法经过 4 次泛洪将所有节点遍历成功,整个过程产生 28 条泛洪消息,3 条是冗余消息.相对于 BFS 算法,UPF 在节省网络资源方面有大幅度提升.

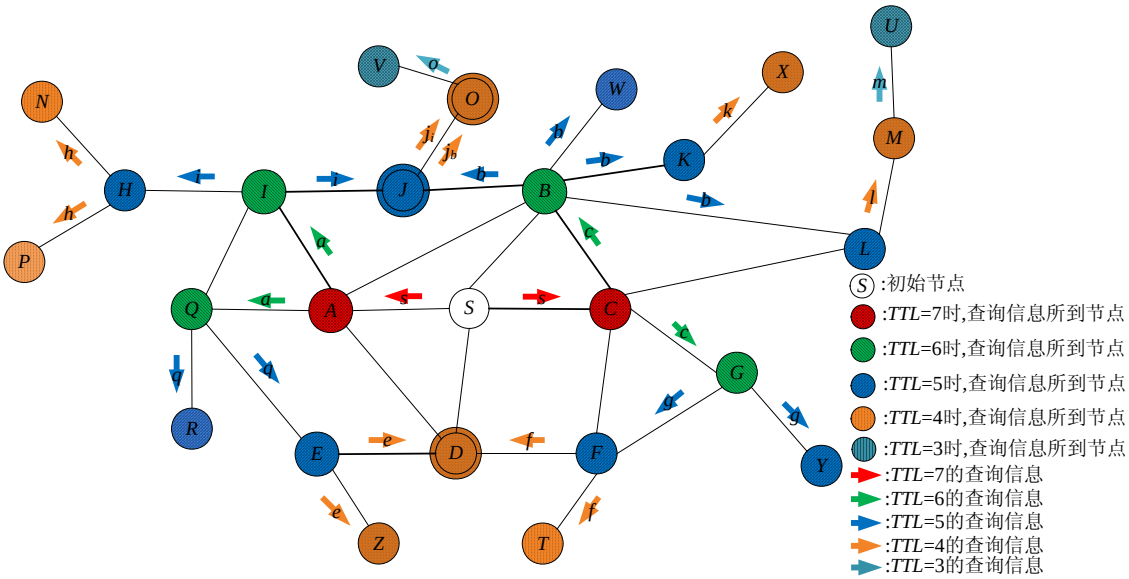


Fig.6 UPF algorithm message routing plan

图 6 UPF 算法消息转发图

3.3.2 BFS 算法和 UPF 算法性能对比分析

在 26 个资源节点中,用户发起资源定位请求,*TTL* 初始值设为 7,BFS 和 UPF 两个算法的查询消息、新接受到消息的节点数量、节点的平均消息接受率、偏覆盖数量和冗余消息数对比见表 1.

Table 1 Performance comparison between BFS algorithm and UPF algorithm

表 1 BFS 算法和 UPF 算法性能对比

Algorithm	*TTL*	Query msg	New peers	Msg per peer	Partial coverage	Redundant information
BFS	7	4	4	1.00	0	0
	6	17	8	2.12	0	3
	5	38	8	4.75	3	26
	4	51	4	12.50	17	47
UPF	7	2	2	1.00	0	0
	6	4	4	1.00	0	0
	5	10	9	1.11	0	1
	4	10	8	0.8	0	2
	3	2	2	1.00	0	0

3.3.3 用户偏好获取开销和隐私保护局限性

用户偏好获取是U2RLS策略进行资源定位的前提.由于用户偏好的多样性和差异性,使得偏好获取开销变成了一个不可忽视的问题.随着时间的推移,用户偏好并非是一成不变的.因此,这需要更为合理的偏好获取方法.在不影响时间和空间复杂度的情况下,U2RLS采用的是一种可拓的关联规则挖掘方法,通过将用户偏好分类,构成具有不同功能属性的用户资源偏好集,结合可拓变原理,推理出新的if-then规则.利用资源分布式存储的动态调整、负载均衡等策略,有效地提高了U2RLS对用户偏好信息的获取、处理能力.在此过程中,U2RLS开销主要集中在兴趣偏好的搜集和处理上:将用户偏好信息数据分成若干属性的集合,确定条件属性集$C=(c_1,c_2,c_3,\ldots,c_m)$和结论属性集$D=(d_1,d_2,d_3,\ldots,d_m)$.U2RLS的额外开销是:计算支持度$sp(C\Rightarrow D)$和置信度$cn(C\Rightarrow D)$.此外,用户的隐私和安全性问题是用户偏好获取时的一个制约条件,U2RLS策略为了定位用户所需的资源,必须记录并分析用户的信息、行为、位置等;但出于隐私和信息安全的考虑,用户不愿意提供完整的准确信息.但利用机器学习和数据挖掘等经典算法,可以挖掘分析出用户偏好轨迹等部分用户信息,同时也可以避开如用户访问

ID、用户访问所在位置、用户生活习惯等威胁用户隐私的信息.

3.4 *TTL*迭代深入策略

TTL 的选取对于 U2RLS 来说是至关重要,查询节点和资源节点之间没有相互的反馈信息,因此对于查询节点来说,如何使得查询合理地进行,完全由 *TTL* 值来控制.*n* 是最大搜索路径,迭代深入算法的优点是:能够避免当 *TTL* 选取不合适时花费在中继节点上的泛洪时间,从而提高资源定位效率.

算法 3. 迭代深入算法.

Input: the set of initial *TTL* $P=\{a,b,c,\ldots,m,n\}$;

Output: the set of complete *TTL* as next time P'.

1. $TTL=a$; do UPF algorithm;
2. **for** each *TTL* value a {
3. **if** ($TTL=0$ && all nodes has been covered) **then** (end the for loop);
4. **else** ($TTL=P_j-P_i$ && do UPF algorithm in a distance P_j-P_i deyond S); //$j=i+1$
5. **if** ($TTL=0$ && all nodes has been covered) **then** (end the for loop about this time);}

3.5 节点离线应对策略

当节点退出或因外部原因离线时,U2RLS 算法采取节点离线应对策略:与该离线节点相关的资源节点,重新计算与该节点的兴趣概率值,更新资源节点兴趣概率树,下一个兴趣子网划分周期来临时,兴趣子网划分算法将与该节点资源相关的节点群重新划分到新的资源子网中.算法 4 描述了整个应对策略.

算法 4. 节点离线应对算法.

Input: the set of old resource nodes: $PG(S)=\{A_1,A_2,A_3,\ldots,A_n\}$, the set of new resource nodes: $PG'(S)=\{\overline{A}_1,\overline{A}_2,\ \overline{A}_3,...,\overline{A}_n\}$, the set of S's old interest subnet nodes: $ISN(S)=\{A'_1,A'_2,A'_3,...,A'_n\}$;

Output: the set of S's new interest subnet nodes $ISN'(S)=\{A''_1,A''_2,A''_3,...,A''_n\}$.

1. **for** each node $A_i\notin PG'(S)$ { //if A_i has been off-line node already, A_i does not belong to $PG'(S)$ definitely.
2. **if** (each node $A_j\in PG(S)$ has the relationship with A_i)
3. **then for** each node A_j { //A_j has the relationship with A_j
4. $\mu=D(A_j||A_i)=\infty$; //A_i is an off-line node, there is no link between A_i and A_j, $D(A_j||A_i)=\infty$.
5. **if** ($A_i\in ISN(S)$) **then** delete A_i from $ISN(S)$;
6. update the interest probability tree;
7. **else** break; }
8. wait for the next user interest subnet period; //user interest subnet algorithm will be reactivated.
9. update the new interest subnet nodes set of $S: ISN'(S)=\{A''_1,A''_2,A''_3,...,A''_n\}$.

4 模拟实验及对比分析

本实验通过Peersim平台扩展其源代码进行模拟,所涉及的网络拓扑图包括Power law随机图、Bimodal拓扑图.比较其他非结构化P2P资源定位方法有APS(adaptive probabilistic search)[21],RW(random walk)[7], BFS[14]和DFS[10].根据P2P网络运行的需求,设计了模拟实验方案和对比分析策略,即对比其他 4 种非结构化P2P资源定位策略:

(1) 根据不同用户需求ω_t,U2RLS策略混合度量周期泛洪函数值的对比分析;
(2) Power law 随机图中,U2RLS 策略是否可以对用户所需资源进行快速、准确的定位搜索;
(3) Bimodal 拓扑图中,U2RLS 策略是否可以大幅度节约网络带宽、减少无必要的网络开销;
(4) Bimodal 拓扑图中,U2RLS 策略的 P2P 网络资源定位过程是否具有时间响应短、查询效率高等特点;
(5) Power law 随机图中,U2RLS 策略是否可以有效避免网络资源定位搜索偏覆盖现象.

4.1 实验模拟参数及默认值

模拟参数及默认值见表 2.

Table 2 Simulation parameters and default values

表 2 模拟参数及默认值

Simulation parameters	Default values
P2P simulator	Peersim
Graph model	Power law/Bimodal
The number of nodes	10 000
The average node degree	8
The number of query requests	1 000
The number of objects	25~800
TTL set	TTL={10,20,30,…,100}
Replication ratio	0.25%/1%/8%
Replication distribution	Zipf (α=0.82)
Query distribution	Zipf (β=0.68)
User requirment metrics	$w_1/w_2/w_3$

4.2 实验1:用户需求w_i下混合周期泛洪函数对比分析

首先验证了不同用户需求w_i下,混合度量周期泛洪函数值和泛洪次数TTL之间的关系.当不同用户依据自身对网络指标偏好来定位所需资源时,选择泛洪节点的数量h_i会因为用户偏好的不同而发生变化,实验 3 次调整用户需求度量的网络连通性w_1、用户共享资源数w_2、用户返回资源定位结果数w_3,分别为

$$w_1=0.5, w_2=0.1, w_3=0.1,$$
$$w_1'=0.1, w_2'=0.3, w_3'=0.1,$$
$$w_1''=0.1, w_2''=0.1, w_3''=0.1.$$

由于利用采样对真实值进行估计过程是一个随机过程,为了减少实验数据的随机性,采用重复 50 次实验然后取算数均值的方法.如图 7 所示:U2RLS在每次转发消息时,选择泛洪节点数量h_i(混合度量周期泛洪函数值)的情况:选取 $w_1'=0.1, w_2'=0.3, w_3'=0.1$时,最多能选择的节点个数为 2 548,相对于 BFS 选择转发消息节点数降低 49.98%.产生这种情况的原因是:用户选择共享资源数量的偏好概率要高于其他两种偏好,共享资源数目高的节点占有更高的比例.

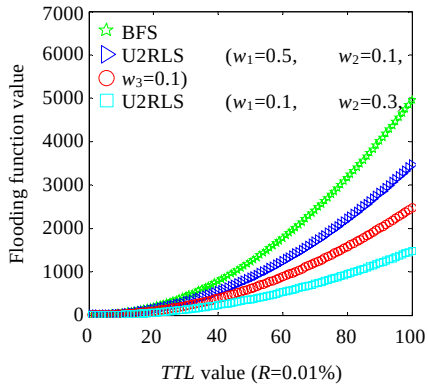


Fig.7 Mixed metric cycle flooding function value vs. TTL value

图 7 混合度量周期泛洪值与 TTL 值

4.3 实验2:资源定位准确率的对比分析

本实验的目的是验证U2RLS的资源定位能力,包括资源定位成功率、资源搜索时间、资源查询命中率.在网络规模是 10 000 节点的条件下,TTL={10,20,30,…,100},取 4 种算法成功率最高的 7 次泛洪,在Power law随机图中,为了模拟现实的网络环境,指数τ设置为 2.1,复制比R分 3 次设置,为R_1=0.25%,R_2=1%,R_3=8%,即资源节点数

分别是 25,100,800.用户需求度量分别为网络连通性w_1=0.2、用户共享资源数w_2=0.1、用户返回资源定位结果数w_3=0.5.

由图 8 可知,当 R=0.25%,TTL=4 时,BFS 查询成功率到达峰值.由于网络中充斥大量泛洪查询信息,当 TTL=1 时,导致节点由于能量的耗尽而大批死亡,BFS 的定位成功率下降近 20%.由于 U2RLS 针对用户偏好进行兴趣子网划分,仅在子网中定位用户所需资源,故有较好的查询成功率,最高为 75.26%,并且随着 TTL 值的减少单调递增.图 8(b)、图 8(c)是复制比 R=1%,R=8%的两种情况.U2RLS 相对于其他 3 种算法,资源定位成功率最高,尤其在资源稀缺的情况下,U2RLS 仍能保持较理想的资源定位成功率.

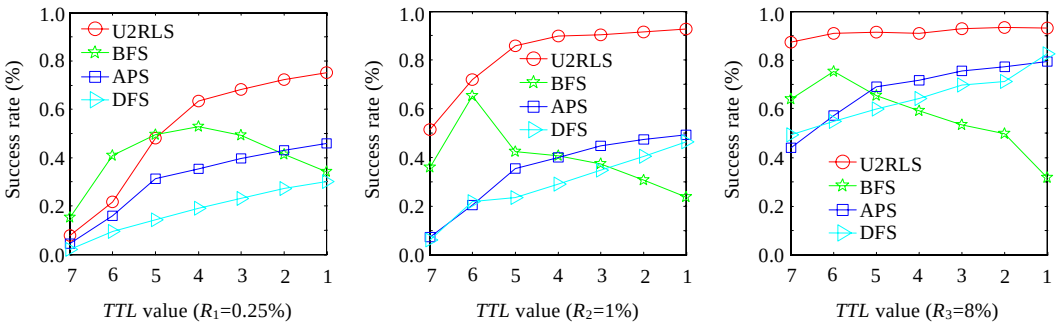


Fig.8 Resource location success rate vs. TTL value in 4 algorithms

图 8 4 种算法的资源定位成功率与 TTL 值

由图 9 可知:4 种算法中,即使在资源较为稀缺的情况下,U2RLS 的查询命中率也处于较高水平;当复制比 R=0.25%时,相对于其他 3 种算法,查询命中率提高将近 30%.这是因为 U2RLS 在用户需求启发下进行泛洪消息,由原来的广度优先搜索策略,演化为带有用户需求的启发式定位策略;同时,摒弃对资源定位失败的消息,抑制了网络资源的消耗和信息冗余的产生.随着复制比在网络中的增大,U2RLS 的优势越发明显.相对于其余 3 种算法,资源查询点击率有大幅度提升,相对 BFS,APS,DFS,U2RLS 提高到 80%左右.

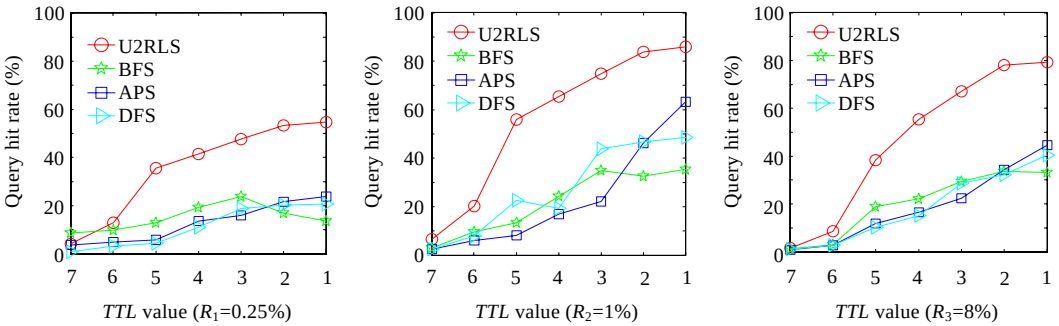


Fig.9 Resource query hit rate vs. TTL value in 4 algorithms

图 9 4 种算法的资源查询命中率与 TTL 值

4.4 实验3:资源定位网络代价开销对比分析

在Bimodal拓扑网络中,实验TTL={10,20,30,…,100},从中选取 25 次迭代结果,用户需求度量为用户共享资源数w_2=0.5、用户资源查询跳数w_4=0.1、查询命中返回时间w_5=0.2.超级节点的个数为 200 个.其中,T_1={A_1,A_2,

$A_3,\ldots,A_{130}\}$,$T_2=\{B_1,B_2,B_3,\ldots,B_{70}\}$;$T_1$的邻居节点的总数$N_1$=1170,$T_2$的邻居节点总数$N_2$=288;$T_1$中节点的平均度数为 9,$T_2$中的邻居节点数为 4.选取$T_1$中超级节点的概率$P_{ultra}$=0.3,选取$T_2$中超级节点的概率$P_{few}$=0.7,复制比$R_1$=0.25%,即采用含有少量邻居节点的查询节点进行泛洪搜索.可以从图 10 知:U2RLS泛洪一次,*TTL*值减少 1,查询冗余消息平均增加 100 条,一次查询产生的冗余信息是BFS的 25%,是APS和DFS的 40%,并随着*TTL*值的减少有增加的趋势,最后趋于稳定.在复制比R=1%和R=8%时,U2RLS的冗余信息相对于其他算法增加的幅度为10%和 15%,而APS和DFS分别增加 28%和 37%.通过调整复制比R,BFS在 3 次实验中产生的冗余消息最多,高达 9 000 余条,并且产生的冗余消息数量没有发生变化.U2RLS相对于其他 3 种算法在产生泛洪冗余消息有明显减少的优势,因此能够大量节省网络资源的开销.

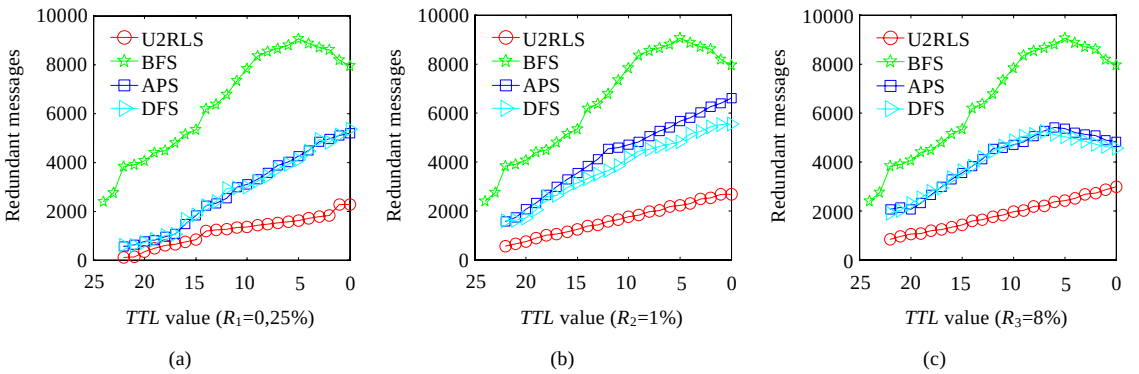


Fig.10 Redundant messages vs. *TTL* value in 4 algorithms

图 10 4 种算法的冗余消息数与 *TTL* 值

将T_1选取超级节点概率P_{ultra}增大为 0.4,T_2选取超级节点的概率P_{few}减小为 0.6,U2RLS其余参数不变,测试其资源定位效率,在副本资源R=0.25%时,如图 11(a)所示:*TTL*∈(25,10),搜索成功率不到 20%;而在*TTL*∈(10,0)的过程中,资源查询效率的一阶导数增加较为明显.造成这种情况的原因是:因为起初 15 次副本定位大多数是在邻居节点较少的T_2中进行的,而副本资源T_1中的概率较大,后 10 次副本定位大多数是在T_1中进行,所以查询效率上升较快,实验中最高为 96.37%,而BFS最高为 58.37%,APS和DFS分别为 23.13%和 38.63%.显然,U2RLS的定位效率要优于其他 3 种算法.在产生的泛洪冗余消息数量方面,U2RLS相对于其他 3 种算法也最小,如图 11(b)、图 11(c),最多不到 2 300 条;而BFS由于无节制的泛洪,导致大批节点能量损耗,使得节点掉线死亡,故在*TTL*=10 时达到峰值,随后逐渐减少,最多达到 8 000 余条冗余信息.因此,U2RLS相对其余 3 种算法节省网络带宽最为突出,从本质上抑制了网络风暴和信息重叠现象的发生.

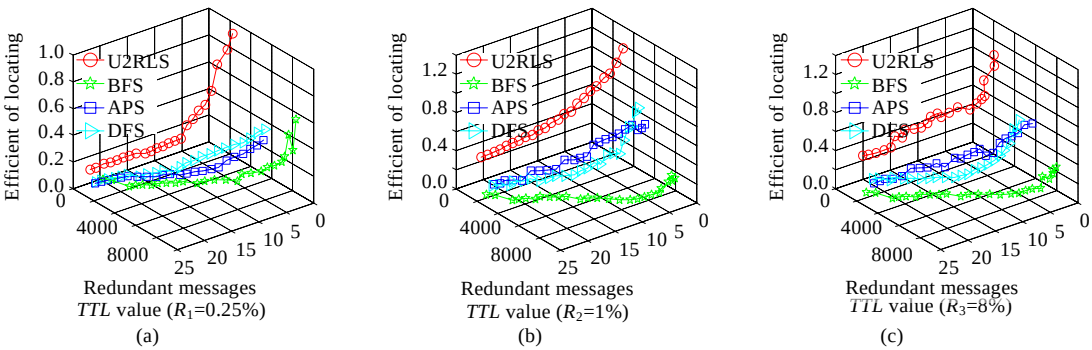


Fig.11 Resource location efficiency vs. redundant messages and *TTL* value in 4 algorithms

图 11 4 种算法的资源定位效率与冗余信息数和 *TTL* 值

在 3 种复制比分别为R_1=0.25%,R_2=1%,R_3=8%的副本比率网络中,网络覆盖度C_h反映了资源定位过程中的查全率.即:C_h越高,覆盖的网络范围就越大,能够搜索定位到节点的数来就越多.由图 12 可知:BFS的覆盖度最高,邻居节点也以此类推.由图 11 也可知:BFS是以消耗极大网络带宽和增加过高的网络负载为代价,可以覆盖掉 10 000 节点中的 96.71%.本文主要是考虑如何避免网络风暴和信息重叠等问题,BFS都满足产生这两种网络缺陷的条件.而U2RLS除了能够以较低的网络代价换取较高的资源定位效果之外,在 4 种算法中,网络覆盖度也有不俗的表现,最高能覆盖到 78.67%的节点.APS和DFS在 3 种复制比设定的情况下,覆盖度劣于U2RLS,因此,资源定位查全率也无法同U2RLS相媲美.

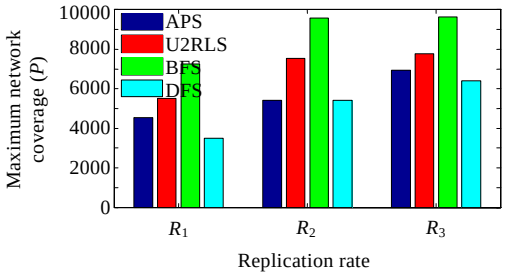


Fig.12 Maximum network coverage vs. replication rate in 4 algorithms

图 12 4 种算法的最大网络覆盖率与复制比

4.5 实验4:资源定位响应时间、存储开销对比分析

在验证用户需求资源定位算法的搜索时间和存储开销上,用户需求度量采用网络连通性w_1=0.2、查询命中返回时间w_5=0.3、网络通信成本w_6=0.1.*TTL*={10,20,30,…,100},随机取其中的 7 次泛洪,在Bimodal网络拓扑结构中,超级节点的个数是 250 个.其中,T_1={A_1,A_2,A_3,…,A_{200}},T_2={B_1,B_2,B_3,…,B_{50}},P_{ultra}取 0.6.由图 13 可知,U2RLS在搜索时间开销上优于其他 3 种算法,尤其在资源稀少的拓扑网络中(R=0.25%,),U2RLS在每个*TTL*值所用的时间都要远小于原BFS近 30%.用户需求度量中有网络连通性w_1和查询命中返回时间w_5这两项,因此在搜索时间上,带有这两项的U2RLS是所有用户需求U2RLS中搜索时间最短的.带有用户需求度量w_1,w_5,w_6的U2RLS为了更好地获取用户所需资源,实验中的平均泛洪节点数为 37.在此消耗的时间是值得的.

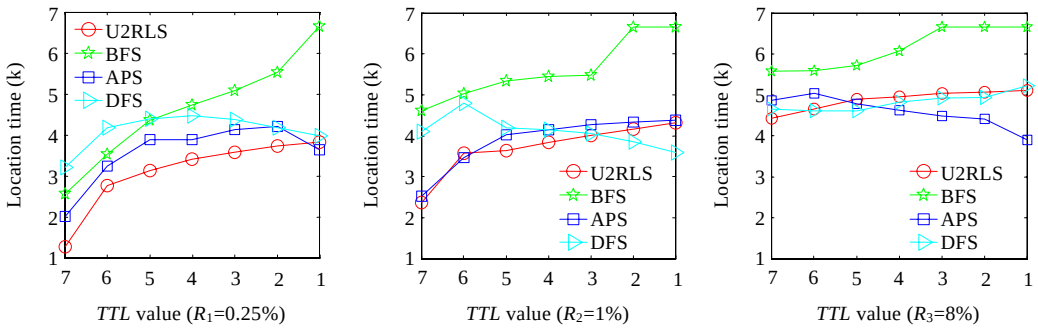


Fig.13 Resource location time vs. *TTL* value of 4 algorithms

图 13 4 种算法的资源定位时间与 *TTL* 值

为了更进一步说明U2RLS响应时间在其他算法中的优势,引入随机漫步算法RW,在P_{ultra}=0.6 时,将 5 种算法的 25 次资源成功定位响应时间做了对比.由图 14(a)所知:RW等待命中的响应时间最长,最短时间为 7.1;而DFS在单次搜索响应中具有绝对优势,最短时间为 0.2;带有用户需求度量w_1,w_5,w_6的U2RLS最快为 1.3,由于在 10

000 节点中泛洪,搜索响应时间$RT\in(0,10)$.U2RLS虽不是最短,但已经处在比较好的水平.随着复制比R的增大,如图 14(b)所示:在搜索响应时间上,U2RLS和DFS几乎重合.而在图 14(c)中,用户需求度量U2RLS的优势再次突显出来,搜索响应时间小于DFS,平均值为 3.3.

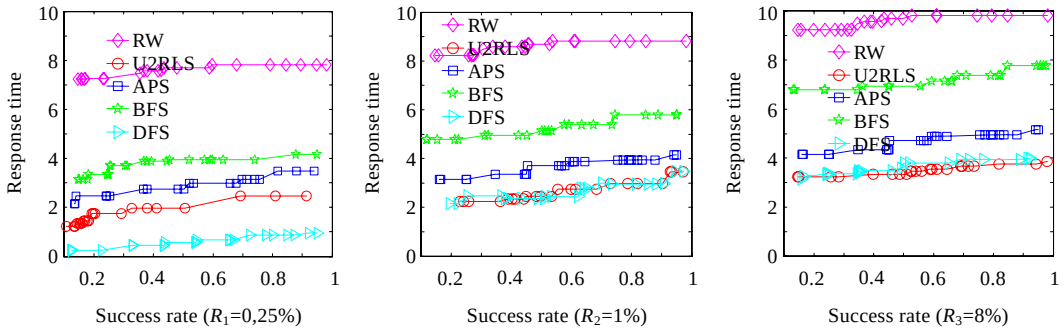


Fig.14 Response time vs. success rate of 5 algorithms

图 14 5 种算法的资源响应时间与成功率

为说明 U2RLS 在存储开销上的优势,本实验将 5 种算法存储时间分成 100 个时间片,并观察每个时间段,所有节点的消息总存储开销,1 个 Time slice 为 1 个单位.

由图 15(a)所知:RW在每个时间片上的总存储开销呈线性函数增长,并随着时间片的增加单调递增;而BFS在 60 个时间片之后的存储字节增加速度变缓.这是由于拓扑网中存在着大量掉线节点,每个时间片上的存储字节数相对RW较低.图 15(b)中:在前 20 个Time slice中,BFS增加迅猛,呈对数分布.而调节T_1中邻居节点的数量,由原来的 170 调整为 80,同时调整P_{ultra}=0.3,如图 15(c)所示:在前 20 个Time slice中,平均存储开销字节相对图 15(b)降低 35%;在存储开销上,U2RLS高于DFS近 20%,虽不是最好,但已经是 5 中算法中排名靠前,算法的时间和空间的复杂度也较低.

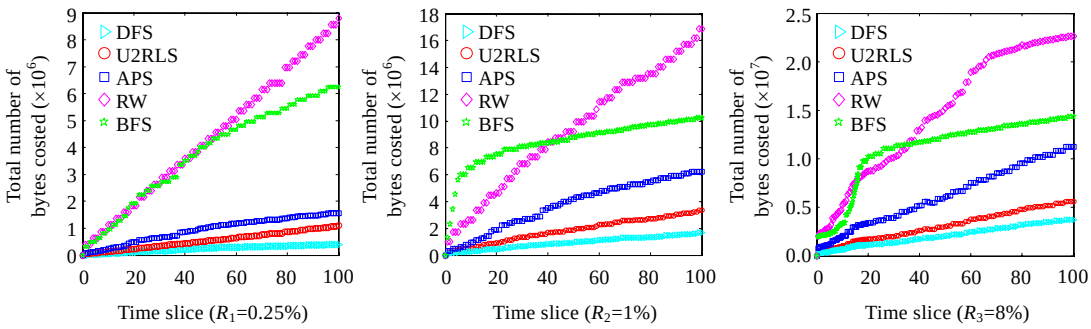


Fig.15 Total number of bytes costed vs. time slice

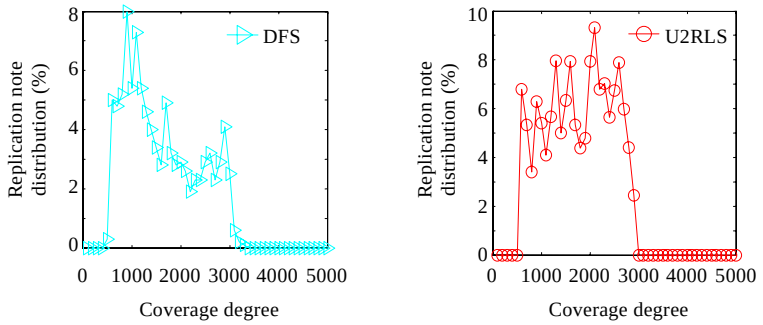
图 15 5 种算法的消息存储开销与时间片

4.6 实验5:资源定位搜索偏覆盖问题解决对比分析

本实验在Power law随机图中进行,选定 10 000 节点的网络环境,节点度数服从τ的指数分布;复制比R=8%,副本服从zipf分布α=0.82;查询服从zipf分布β=0.68.再次选取用户度量,网络连通性w_1=0.3,查询返回定位结果数量w_5=0.2,网络通信成本w_6=0.2.

(1) 当选用 1 个中继节点泛洪查询消息时,TTL最大值设为 100,复制比R=8%.将 5 种算法的覆盖度和副本覆

盖数量占总体节点数量的百分比情况进行对比分析:图 16 在 5 种算法中选出 2 种较为理想的算法副本节点分布和覆盖度的对比分析.从图 16(a)可知:DFS在进行资源定位搜索时,在覆盖度C_h∈(0,1000)范围内,最大覆盖到副本数量的 8%,副本节点分布在 5%以上的有 6 次;而U2RLS在覆盖度C_h∈(2000,3000)范围内,最大覆盖到副本数量的 9.31%,副本节点分布在 5%以上的有 17 次,其余算法无法达到这个水平.

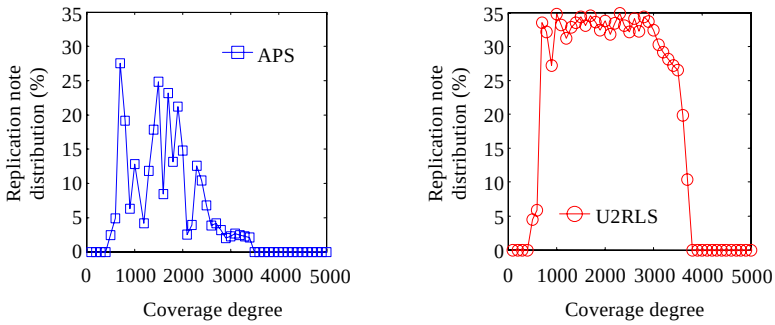


(a) (b)

Fig.16 Replication note distribution vs. coverage degree in DFS & U2RLS

图 16 DFS & U2RLS 的副本节点分布率与覆盖度

(2) 当选用 10 个中继节点同时泛洪查询消息时,复制比R=8%,将 5 种算法的覆盖度和副本分布数量再次进行对比分析,同样选出 2 种副本分布率较为理想的情况.由图 17 可知:U2RLS优势明显,5 000 节点最大能够覆盖副本数量的 35.23%,并且C_h∈(1000,3000)时,副本最为集中 25%以上有 29 次;而APS最大副本节点分布率为 27.3%,25%以上的只有 1 次.相比之下,U2RLS的副本分布范围更广,搜索节点的数量更多.

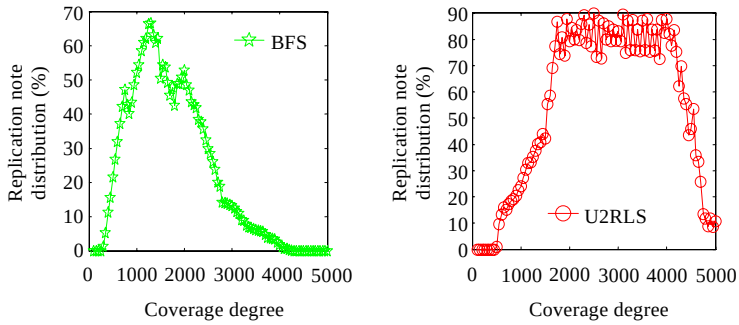


(a) (b)

Fig.17 Replication note distribution vs. coverage degree in APS & U2RLS

图 17 APS & U2RLS 的副本节点分布率与覆盖度

(3) 当选用 100 个中继节点同时泛洪消息时,*TTL* 最大值设置为 100,复制比 R=8%,将 5 种算法的覆盖度和副本分布进行第 3 次对比分析.如图 18 所示:在 5 种算法中,第 3 次选出 2 种较为理想的副本分布情况.由图 18(a)可知:BFS 的副本分布率最多能够达到 66.5%,在 1 460 个节点中,二阶导数达到峰值,出现拐点,虽然分布较高,但此种情形过为稀少,由于 BFS 采用广度优先搜索策略,泛洪次数增大到一定程度,副本分布范围最广,但产生的偏覆盖现象最为严重.由图 18(b)得知:U2RLS 的节点覆盖率集中于 2 个区间,分别为 C_{h_1} ∈(500,1700), C_{h_2} ∈(1900,4300)高值分别为 46.2%和 89.8%,次数各为 20 次和 51 次.显然,U2RLS 能够覆盖更多副本.



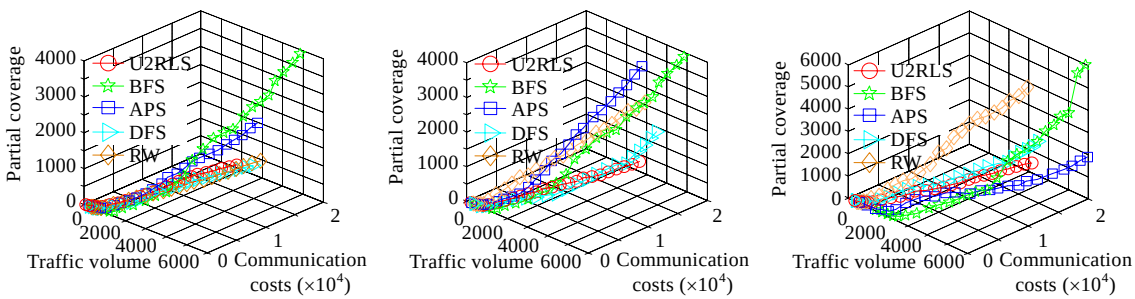
(a) (b)

Fig.18 Replication note distribution vs. coverage degree in BFS & U2RLS

图 18 BFS & U2RLS 的副本节点分布率与覆盖度

搜索偏覆盖实验采用的是将 5 种算法产生的偏覆盖数量、网络负载量、通信成本消耗量进行三维图像汇总,如图 19 所示.通过图像对比分析,在复制比R_1=0.25%情况下,U2RLS和DFS产生的偏覆盖数量较少,分别是259和85,而BFS产生的网络偏覆盖为3 864,U2RLS在避免偏覆盖方面具有绝对优势.随着副本在网络中的增加,复制比R_2=1%时,5 种算法的网络负载量和通信成本消耗也普遍升高,在偏覆盖数方面,APS上升较为明显.当复制比R_3=8%时,RW和BFS的偏覆盖情况较为严重,分别为 3 169 和 5 964.

由此可知:U2RLS 采用了混合度量周期性泛洪策略,它的偏覆盖数、网络负载量,通信成本消耗情况,相对于其余 4 种算法,仍然处在一个较低水平.



(a) R_1=0.25% (b) R_2=1% (c) R_3=8%

Fig.19 Partial coverage vs. traffic volume and communication cost

图 19 偏覆盖数与交通量和网络开销

5 结束语

非结构化 P2P 网络资源定位搜索具有很高的盲目性和不确定性,搜索过程中产生大量的通信负载和信息重叠现象,造成严重的网路带宽负担以及巨大的索引维护开销,进而影响了用户在资源定位时的查准率、查全率.本文针对传统的 P2P 网络资源定位泛洪策略的缺陷,提出了面向用户需求的非结构化 P2P 资源定位泛洪策略,该策略将用户需求、用户偏好、用户兴趣融入其中.从理论上证明了泛洪策略极易出现网络风暴、信息重叠和搜索偏覆盖问题,并根据泛洪策略的特征,将用户资源所在节点群进行子网划分,利用混合度量周期泛洪策略,选择性地转发用户需求查询信息.为降低消息的过度泛洪,导致节点能量耗尽而死亡的可能性,本文改进了传统泛洪策略中 *TTL* 的计算方式,提出了 *TTL* 迭代深入策略,抑制了网络资源过度消耗等问题.最后,通过模拟实验验证了 U2RLS 策略具有搜索成功率高、网络资源消耗少、查询响应时间短、网络覆盖范围广等优势.

References:

[1] Li DS. Research on peer-to-peer resource location in large-scale distributed systems [Ph.D. Thesis]. Changsha: National University of Defense Technology, 2005 (in Chinese with English abstract).

[2] Cheema AS, Muhammad M, Gupta I. Peer-to-Peer discovery of computational resources for grid applications. In: Proc. of the IEEE ACM Int'l Workshop Grid Comput. IEEE Computer Society, 2005. 179−185. [doi: 10.1109/GRID.2005.1542740]

[3] Zhao Q, Liu JY, Xu JD. Improving search on gnutella-like P2P systems. In: Proc. of the 7th Int'l Conf. on Computational Science (ICCS 2007). Heidelberg: Springer-Verlag, 2007. 877−880. [doi: 10.1007/978-3-540-72590-9_133]

[4] Huang YS, Meng XW, Zhang YJ. Strategy of content location of P2P based on the social network. Ruan Jian Xue Bao/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]

[5] Ma WM, Meng XW, Zhang YJ. Bidirectional random walk search mechanism for unstrucured P2P network. Ruan Jian Xue Bao/ Journal of Software, 2012,23(4):894−911 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4086.htm [doi: 10.3724/SP.J.1001.2012.04086]

[6] Kim H, Kim Y, Kim K, Kang S. Restricted path flooding scheme in distributed P2P overlay networks. In: Proc. of the Int'l Conf. on Information Science and Security (ICISS 2008). New Jersey: IEEE Computer Society, 2007. 58−61. [doi: 10.1109/ICISS. 2008.43]

[7] Li ZT, Yu WN, Liu G. DPFSL: Decreasing probability flooding algorithm in P2P network. Journal on Communications, 2006,27(11A):246−250 (in Chinese with English abstract).

[8] Nourazar F, Sabaei M. DAPF: An efficient flooding algorithm for mobile ad-hoc networks. In: Proc. of the 2009 Int'l Conf. on Signal Processing Systems (ICSPS). IEEE Computer Society, 2009. 594−598. [doi: 10.1109/ICSPS.2009.112]

[9] Lou JQ, Zhou SJ, Wu CJ, Deng YY, Yang XQ. Adaptive flooding routing algorithm in unstructured P2P. In: Proc. of the 2006 Int'l Conf. on Communications, Circuits and Systems (ICCCAS). IEEE Computer, 2006. 1557−1561. [doi: 10.1109/ ICCCAS.2006. 284968]

[10] Sanderson P. Identifying an existing file via Kazaa artefacts. Digital Investigation, 2006,3(3):174−180. [doi: 10.1016/j.diin.2006.08. 011]

[11] Zhang KL, Wang S. LinkNet: A new approach for searching in a large peer-to-peer system. Chinese Journal of Computer, 2006,29(4):611−617. [doi: 10.1007/978-3-540-31849-1_24]

[12] Feng GF, Mao YC, Lu SL, Chen DX. PeerRank: A strategy for resource discovery in unstructured P2P systems. Ruan Jian Xue Bao/Journal of Software, 2006,17(5):1098−1106 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/1098.htm [doi: 10.1360/jos171098]

[13] Johnson DL, Pejovic V, Belding EM, Van SG. VillageShare: Facilitating content generation and sharing in rural networks. In: Proc. of the 2nd ACM Symp. on Computing for Development (DEV 2012). New York: ACM Press, 2012. [doi: 10.1145/2160601. 2160611]

[14] Chen Z, Guo SZ, Yang YX, Zheng KF. Research of the apriority policy-based multi-hop BFS search algorithm in P2P network. In: Proc. of the 2007 IFIP Int'l Conf. on Network and Parallel Computing Workshops. IEEE Computer Society, 2007. 471−476. [doi: 10.1109/ICNPCW.2007.4351529]

[15] Adamic LA, Lukose RM, Punlyani AR, Huberman BA. Search in power-law networks. Physical Review E—Statistical, Nonlinear, and Soft Matter Physics, 2001,64(4II):461351−461358. [doi: 10.1103/PhysRevE.64.046135]

[16] Zhang CH, Qiu XF, Ji Y. Comprehensive Analysis of P2P Technology. Beijing: Posts & Telecom Press, 2010 (in Chinese).

[17] Aiello W, Chung F, Lu L. A random graph model for massive graphs. In: Proc. of the Conf. on Annual ACM Symp. on Theory of Computing. New York: ACM Press, 2000. 171−180. [doi: 10.1145/335305.335326]

[18] Zhuang ZY, Liu YH, Lionel MN. Hybrid periodical flooding in unstructured peer-to-peer networks. In: Proc. of the 2003 Int'l Conf. on Parallel Processing. 2003. 171−178. [doi: 10.1109/ICPP.2003.1240578]

[19] Conesa J, Caballe S, Ganan D, Prieto J. Towards an ontology to model and represent collaborative learning data. In: Proc. of the Communications in Computer and Information Science. 2013. 351−356. [doi: 10.1007/978-3-642-35879-1_42]

[20] Guo XM. The research on sharing file consistency maintencance in unstructured P2P system [MS. Thesis]. Changsha: Hunan University, 2008 (in Chinese with English abstract).

[21] Dimitrios T, Nick R. Adaptive probabilistic search (APS) for peer-to-peer networks. In: Proc. of the Int'l Conf. on Information and Knowledge Management. New York: ACM Press, 2008.

附中文参考文献:

[1] 李东升.基于对等模式的资源定位技术研究[博士学位论文].长沙:国防科学技术大学,2005.

[4] 黄永生,孟祥武,张玉洁.基于社会化网路特征的 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]

[5] 马文明,孟祥武,张玉洁.面向非结构化 P2P 网络的双向随机漫步搜索机制.软件学报,2012,23(4):894−911. http://www.jos.org.cn/1000-9825/4086.htm [doi: 10.3724/SP.J.1001.2012.04086]

[7] 李之棠,余万能,刘刚.DPFSL:P2P 网络的递减概率泛洪算法.通信学报,2006,27(11A):246−250.

[12] 冯国富,毛莺池,陆桑璐,陈道蓄.PeerRank:一种无结构 P2P 资源发现策略.软件学报,2006,17(5):1098−1106. http://www.jos.org.cn/1000-9825/17/1098.htm [doi: 10.1360/jos171098]

[16] 张春红,裘晓峰,弭伟,纪阳.P2P 技术全面解析.北京:人民邮电出版社,2010.

[20] 郭晓梅.非结构化 P2P 网络资源一致性维护算法研究[硕士学位论文].长沙:湖南大学,2008.

何明(1982－),男,辽宁丹东人,博士生,主要研究领域为 CDN-P2P 技术,数据挖掘,网络服务,通信软件.

张玉洁(1969－),女,副教授,主要研究领域为智能信息处理,通信软件,网络服务.

孟祥武(1966－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为网络服务,通信软件,人工智能.