一种基于响应追踪的非结构化 P2P 资源查找机制

王淑玲 杨寿保* 王喜妹 武 斌
(中国科学技术大学计算机科学与技术学院 合肥 230026)

摘 要：为了提高非结构化 P2P 中资源查找的效率，针对查找请求的关键字重复出现以及资源共享不平均的现象，提出了基于响应追踪的资源查找机制 RTRSM (Response Tracing based Resource Searching Mechanism)，对请求响应的内容以及响应的来源进行追踪，并记录历史信息，以此构建提示型资源查找。在查找请求转发阶段，依据查找请求与历史记录的相关程度，自适应地调整转发策略，在保证较高查找成功率的同时，降低查找代价，提高查找效率。仿真实验表明，基于响应追踪的资源查找机制 RTRSM 可扩展性较好，能以较小的开销扩展到较大规模的系统中去。仿真实验还表明，RTRSM 能保证 90%以上的查找成功率。与洪泛法相比，RTRSM 能减少 54.2%的平均消息数，减少 56.4%的平均跳数；与 SQR (Scalble Query Routing)相比能减少 28.9%的平均消息数，减少 6.4%的平均跳数。

关键词：非结构化 P2P 网络；响应追踪；资源查找；布鲁姆过滤器；提示型资源查找
中图分类号：TP393 **文献标识码**： A **文章编号**：1009-5896(2011)08-1937-06
DOI: 10.3724/SP.J.1146.2010.00736

Response Tracing Based Resource Searching Mechanism for Unstructured P2P Network

Wang Shu-ling Yang Shou-bao Wang Xi-mei Wu Bin
(*School of Computer Science*, *University of Science and Technology of China*, *Hefei* 230026, *China*)

Abstract: There are some instinctive characteristics in unstructured P2P network, such as frequent reappearance of keywords and uneven resource sharing. To enhance performance of resource searching, a Response Tracing based Resource Searching Mechanism (RTRSM) is proposed, which traces the content and the latest source of responses to construct an informed search. RTRSM adaptively forwards the query to the suitable neighbor, according to the related degree of the storing information and the query itself. By constructing an informed search, RTRSM hopes to maintain high success rate, decrease searching cost and promote searching efficiency. Simulation results show that RTRSM is scalable. It can be extended to large scale unstructured P2P network with a low cost. Simulation results also show that RTRSM achieves good searching efficiency. RTRSM decreases 54.2% of average message counts and 56.4% of average hops by contrast to flooding, and decreases 28.9% of average message counts and 6.4% of average hops by contrast to SQR (Scalable Query Routing), while the success rate keeps up to 90%.

Key words: Unstructured P2P network; Response tracing; Resource searching; Bloom filter; Informed search

1 引言

近年来，随着分布式文件共享系统的发展，P2P 网络模式得到了越来越多的关注，尤其是以 Gnutella[1]为代表的非结构化 P2P 网络。非结构化 P2P 网络结构简单，节点以任意形式进行组织，但正是其简单性给该网络模式下的资源查找带来了极大的挑战。

在非结构化 P2P 网络中，依照节点是否保存资源的定位信息可将资源的查找机制划分为两类：盲目型资源查找和提示型资源查找 (informed search)[2]。

洪泛法[3]属于盲目型资源查找，这类资源查找机制不需要在节点中保存与资源定位相关的信息，转发过程较为盲目，容易产生大量的冗余信息，增加网络负载。

提示型资源查找[2,4−7]需要在节点中保存与资源定位相关的信息，指引资源查找请求的转发。此类查找机制充分利用了启发式搜索的思想，对于每一个资源查找请求并不保证返回最好的响应，但总是返回可以接受的结果。文献[4−7]通过在节点中保存节点拥有的资源信息，并在合适的时机将这些信息

2010-07-12 收到，2011-06-16 改回
国家自然科学基金(60673172)，国家 863 计划项目(006AA01A110)和研究生创新基金资助课题
*通信作者：杨寿保 syang@ustc.edu.cn

共享给其他节点，以此构建提示型资源查找。文献[6]提出了一种基于路由学习的资源转发策略，通过记录历史的资源查找请求的满足情况，指引当前的资源请求转发。文献[2]基于 P2P 网络中节点共享资源以及网络拓扑的不均衡特性，对节点拥有的资源数和节点的度综合考虑，将资源查找请求转发给拥有较多资源并且节点度较大的节点，加快查找速度，提高请求的成功率。

上述的提示型资源查找策略中，所有的节点通过维护自身以及周围邻居的底层信息，如网络拓扑、共享的资源等，构建路由表，而资源分布的不均匀特性导致大部分节点维护的信息不充分，因此不能有效地指引请求的转发。

文献[8,9]指出，在非结构化的 P2P 网络中，资源查找请求中关键字的重复出现次数服从幂率(power-law)分布。因此，当相同或者类似的资源查找请求被提交的时候，可依据历史记录相应地转发查找请求，提高查找效率。文献[10,11]指出，P2P 网络中 70%的节点并不贡献资源，网络中大部分的资源仅由少部分节点贡献，节点的请求只在少数节点上能得到满足。因此在转发资源请求时，将请求转发给响应了较多请求的节点，能达到更高的资源查找效率。

基于上述考虑，本文提出了一种基于响应追踪的资源查找机制(Response Tracing based Resource Searching Mechanism, RTRSM)，对资源查找请求响应进行追踪，并记录有效信息，构建提示型资源查找方法，保证较高的资源查找成功率的同时，降低资源查找的代价，提高资源查找的效率。

本文的其余部分安排如下：第 2 节详细描述 RTRSM 的操作过程，第 3 节对 RTRSM 进行理论分析，第 4 节是模拟实验，第 5 节是结束语。

2 基于响应追踪的资源查找机制

基于响应追踪的资源查找机制 RTRSM 通过追踪资源查找请求的响应来指引查找请求的转发，包括追踪响应的请求内容以及追踪响应的来源。节点在收到查找请求后，评估每个邻居节点的历史记录与查找请求的相关程度，并自适应地调整查找请求的转发策略。

为了适应 RTRSM 的需求，在非结构化 P2P 网络中，每个节点为其所有的邻居节点维护两个表项：通过该邻居节点收到的请求响应的查找信息记录，在本文中表现为查找请求的关键字；通过该邻居节点收到的响应的数。这两部分将在下文详细介绍。

为了便于描述，在表 1 中定义了本文需要用的符号。

2.1 响应追踪

RTRSM 对响应的追踪包括追踪响应所响应的内容以及追踪响应的来源。

表 1 文中所用符号

参数	定义	参数	定义
g_{ij}	节点 i 和节点 j 直接相连，则 g_{ij}=1，否则为 0	G	$(g_{ij})_{n\times n}$
d_i	节点 i 的邻居数	n	节点数
c_{ij}	节点 i 收到的来自于节点 j 的响应数	s	每个查找请求包含的关键字数
q.key	查找请求 q 包含的关键字	m	Bloom Filter 的位数
$\mathrm{Res}_{q,j}^{A}$	查找请求 q 在 j 跳后到达节点 A	h_i	EDBF 的第 i 个哈希函数
C	$(c_{ij})_{n\times n}$	k	EDBF 关联的哈希函数的个数

本文采用一种改进后的 BF(Bloom Filter)结构 EDBF(Exponentially Decaying BF)[7]对请求响应所响应的内容进行追踪。在非结构化 P2P 网络中，每个度为 d 的节点维护由 d 个 **EDBF** 向量组成的信息表，其中每个 **EDBF** 向量记录了该邻居节点的响应所响应的内容的历史记录以及通过该邻居节点可达的其他节点的历史响应内容记录。节点 A 为邻居节点 B 维护 **EDBF** 向量可以表示为

$$\left.\begin{aligned} \boldsymbol{R}_A^B &= \boldsymbol{R}_A^B \bigcup_{1}\{q \mid \mathrm{Res}_{q,j}^{B} \ \Lambda \ \mathrm{Res}_{q,j+1}^{A}\} \bigcup_{1-\alpha} \mathfrak{R}_B \\ \mathfrak{R}_B &= \bigcup_{1} R_B^u \mid u \in \mathrm{neighbour}(B) \end{aligned}\right\} \quad (1)$$

其中 $\boldsymbol{X}\bigcup_{1-\alpha}\boldsymbol{Y}$ 表示对向量 $\boldsymbol{Y}$ 进行衰减因子$1-\alpha$的 BF 的插入操作，即首先对向量 $\boldsymbol{Y}$ 中所有值为 1 的位以概率$1-\alpha$被赋值为 0，以概率α保持不变，然后对 $\boldsymbol{Y}$ 衰减后的向量与向量 $\boldsymbol{X}$ 执行 BF 的插入操作，本文称此操作为衰减因子为$1-\alpha$的 **Union**。节点 A 为邻居节点 B 生成 **EDBF** 向量的算法如表 2 的算法 1 所示，响应的更新过程为 **ReceiveUpdateBF**。

表 2 节点 *A* 为邻居节点 *B* 生成 EDBF 向量的算法

算法 1 节点 A 为邻居节点 B 维护的 **EDBF** 向量的更新过程

```
Union (R_A^B, Res_{q,j}^B)
 if (True (Res_{q,j+1}^A) )
     for all q.key
         for i←1 to k
             j ←Hash( i, q.key )
             R_A^B [j]←1
   endif
ReceiveUpdateBF(ℜ_B)
/*function Union( R_A^B ,ℜ_B) with decay factor 1−α */
for i←1 to m
  if (Random( 1 − α ))
  ℜ_B[i]←0
endfor
R_A^B ← R_A^B ⊙ ℜ_B
```

节点 A 收到一个资源查找请求 q 后，首先将在本地的历史记录中查找，查看是否有与所请求资源类似的资源存在，这一过程是 **EDBF** 中的查询操作，本文中称为 **Search** 操作。请求 q 的每个关键字 $q.\text{key}_j$ 在节点 A 的 $\boldsymbol{R}_A^B$ 向量中查找的返回值为 $\delta_A^B(q.\text{key}_j)$，表示由 k 个哈希函数将 $q.\text{key}_j$ 映射至的 k 个位置中 $\boldsymbol{R}_A^B$ 为 1 的位数，满足表达式(2)。

$$\delta_A^B(q.\text{key}_j)=\left|\left\{i\left|\boldsymbol{R}_A^B\left[h_i(q.\text{key}_j)\right]=1,i=\{1,2,\cdots,k\}\right.\right\}\right| \quad (2)$$

在 P2P 系统中，存在着大量的自私节点，网络中的大部分资源只由少数一部分节点提供，因此大部分的资源查找请求只在少部分节点上得到满足。本文在追踪响应所响应的内容的基础上，追踪响应的来源，以此发掘网络中的热点节点。在资源请求转发阶段，将无相关历史记录的查找请求以较大概率转发至这些热点节点上，以进一步降低查找的代价，提高查找的性能。

基于响应追踪的资源查找机制对响应的来源进行追踪，统计每个邻居节点响应资源请求的数目，并引入了热点度 ω 来表示其热点程度，并按式(3)进行定义。

$$\omega_A^B=c_{AB}\Big/\sum_j c_{Aj},\quad j\in\text{neighbour}(A) \quad (3)$$

2.2 请求转发

RTRSM 机制转发资源查找请求时，首先将查找请求与该节点上所有的 EDBF 向量进行匹配，评估每个邻居节点的历史响应信息记录与该查找请求的相关程度，并由此自适应地调整邻居节点的“身份”。如果某个邻居节点中存在与查找请求相匹配或者较相似的历史记录，则该节点以较大的概率作为相关节点的转发资源，否则将以热点节点的“身份”转发资源。

本文用贡献排名 conRank 来表征每个邻居节点作为两种“身份”的综合能力，即查找请求通过该邻居节点的得到响应的可能性大小，按式(4)进行定义。

$$\begin{aligned}\text{conRank}_A^B(q)=&\left[\sum_{i=1}^{s}\delta_A^B(q.\text{key}_i)\Big/s\right]^2\\&+\left(1-\sum_{i=1}^{s}\delta_A^B(q.\text{key}_i)\Big/s\right)\times\omega_A^B\end{aligned} \quad (4)$$

节点 A 收到资源查找请求 q 后，相应的转发算法如算法 2 所示。

3　算法分析

假设系统中共有 n 个节点，每个节点的邻居信息表中包含有 d 个表项的 **EDBF** 向量，每个向量有 m 位，k 个哈希函数。系统的拓扑图为随机图结构，节点度数在 $[a,b]$ 范围内均匀分布，节点的平均度数为 $d_a=(a+b)/2$。现考量与节点 A 的距离为 i 的节点数。当 i 较小时，可近似认为系统拓扑类似于树形结构(无环)，从而与 A 的距离为 i 的节点数 n_i 约为 $d_a(d_a-1)^{i-1}$。假设关键字的重复次数服从幂率分布。

表 3　算法 2 节点 *A* 为转发查找请求 *q* 的转发算法

```
RequesForwarding(q)
(1)  max ← Null
(2)  for all u∈neighbor(A)
(3)      relate ← Σδ_A^u(q.key_i)/s
(4)      if relate = 0 && ω_A^u < Threshold
(5)          break
(6)      conRank ← relate *relate +(1−relate)* ω_A^u
(7)      if conRank > maxRank
(8)          max ← u
(9)  endfor
(10) if max = Null
(11)     broadcast
(12) else sendto(p, max)
```

3.1 查询请求的重复次数

若关键字的重复次数服从幂率分布，即对于某个关键字，其出现 x 次的概率为

$$p(x)=(\xi-1)x^{-\xi} \quad (5)$$

若一个包含了 s 个关键字的资源查找请求重复出现时，s 个关键字均重复出现了至少两次，故一个查找请求的重复出现概率为

$$p_{\text{repeat}}=\int_2^\infty\cdots\int_2^\infty\prod_{i=1}^{s}\frac{p(x_i)}{x_i}\,\mathrm{d}x_1\mathrm{d}x_2\cdots\mathrm{d}x_s=2^{s(1-\xi)} \quad (6)$$

3.2 命中率

节点收到一个资源查找请求后，首先将节点中所有邻居节点的响应历史信息与查找请求进行匹配。如果在某个邻居节点中有相关或相似的记录，或者存在有热点节点，则称该查找请求在该节点上命中，现对这一概率进行分析。

首先分析 A 的每个 **EDBF** 向量中任意一位为 0 的概率。令所有可能引起 A 的响应信息发生变化的位数的期望为 S_i，与 A 距离为 j 的每个节点中，每次 **EDBF** 向量的改变至多有 k 位值为 0 的位变成 1，并且以概率 α^j 传送至 A 的某个邻居的 **EDBF** 向量中，因此 S_i 满足以下表达式：

$$S_i=n_1k\times\alpha+n_2k\times\alpha^2+\cdots=\frac{k\alpha\times d_a}{1-\alpha\times d_a} \quad (7)$$

A 的每一个 **EDBF** 向量中任意一位为 0 的概率以及任意一位为 1 的概率分别由 p_0 和 p_1 表示，若 **EDBF** 向量中的某一位在一次更新结束后，仍然为

0，则所有可能引起该位发生变化的更新均未引起其值的变化，否则这一位将在更新结束后变为 1，因此 p_0和 p_1满足以下表达式：

$$\left.\begin{aligned} p_0 &= (1-1/m)^{S_i/d_a} \approx e^{-\frac{k\alpha}{(1-\alpha\times d_a)m}} \\ p_1 &= 1-(1-1/m)^{S_i/d_a} \approx 1-e^{-\frac{k\alpha}{(1-\alpha\times d_a)m}} \end{aligned}\right\} \tag{8}$$

如果节点 A 的某个邻居 B 的响应历史记录中存在着相关记录，则查找请求 q 中至少存在关键字 $q.\text{key}_i$，并满足由 k 个哈希函数将 $q.\text{key}_i$ 映射至的 k 位中，$\boldsymbol{R}_A^B$ 至少有一位为 1。在这 k 位中，$\boldsymbol{R}_A^B$ 有 l 位为 1 的概率为

$$\text{sim}_i(l) = p(\delta_A^B(q.\text{key}_i) = l) = C_k^l \cdot p_1^l \cdot p_0^{k-l} \tag{9}$$

现在考虑 A 的一个邻居节点 B 成为 A 的所有邻居节点中的热点节点的概率。节点 B 若要成为 A 的所有邻居中的热点节点，则需要满足：

$$\omega_A^B \geq \eta \tag{10}$$

其中 η 为设置的阈值，即若节点 A 共响应了 T 个请求，则至少有 $t = T\times\eta$ 个查找响应来自于 B，此概率为

$$ph_A^B(\eta) = p(\omega_A^B > \eta) = \frac{1}{d_a}\sum_{j=0}^{T-t} C_{d_a+j-2}^j \Big/ C_{d_a+T-1}^T \tag{11}$$

综上所述，查找请求 q 在节点中能够被命中的概率为

$$\text{hit} = 1-\text{sim}_i(0)^{s\cdot d_a}\times(1-ph(\eta))^{d_a} \tag{12}$$

3.3 平均消息数

在 RTRSM 机制中，节点收到查找请求 q 后，将评估请求 q 转发给各个邻居节点后能够被响应的可能性大小。如果存在某个邻居节点有足够的信息指引请求 q 的转发并以较大概率能返回响应，则将 q 转发给该节点，否则广播此查找请求。在 RTRSM 中由历史记录与请求 q 的相关程度自适应地调整转发策略，在此，假设查找请求向热点节点转发与向相关节点转发互不影响，即如果某邻居节点的历史信息记录与请求相关程度较高，则不管该邻居节点是否是热点节点，资源查找请求均被转发至该节点。反之，若一个节点是热点节点，则无论该节点是否是相关节点，查找请求均被转发至该节点。

一个未被响应过的查找请求在其转发的每一跳时，如果遇见了热点节点，则只向热点节点转发，否则广播。其引发的消息数的期望为

$$E_0 = (1-\text{hit})\times d_a + \text{hit}\times\text{sim}_i^{s\cdot d_a}(0)\times ph(\eta) \tag{13}$$

一个曾经被响应过的查找请求在其转发的过程中，遇见了有相似或相同记录的节点，则其引发的消息数的期望为

$$E = \sum_{l_1,l_2\cdots l_s}\prod_{i=1}^{s}\text{sim}_i(l_i)\times\log_\alpha\left(ks\Big/\sum_{i=1}^{s}l_i\right) \tag{14}$$

因此，对于一个查找请求来说，其引发的平均消息数为

$$\text{MsgC}=p_{\text{succ}}\left[\frac{1}{\text{TTL}}\times\sum_{j=1}^{\text{TTL}}\frac{n_j}{N}E_0^j+E\times(Eq-1)\right]\Big/Eq + (1-p_{\text{succ}})E_0^{\text{TTL}} \tag{15}$$

$$Eq = \text{queryNum}\times p_{\text{repeat}} \tag{16}$$

其中 p_{succ} 为查找请求的成功率，queryNum 为查找请求的总数。

取定 ξ =2.3, k=13, d_a=4, s=1, m=96k, α = 0.2, η =0.5, queryNum=5000 时，MsgC=41.7847，实验值为 34.778，理论值与实验值偏差 20.15%。偏差的原因在于理论分析所基于的假设：请求向相关节点转发与向热点节点转发互不影响。

4 模拟验证

本文仿真实现了基于响应追踪的资源查找机制 RTRSM，并与 Gnutella 的洪泛法以及文献[7]的 SQR 在查找成功率、平均消息数和平均跳数方面进行了比较分析。

4.1 模拟环境

在本文的模拟实验中，使用 Brite[12]作为拓扑产生器生成一层 Router-only 拓扑结构。拓扑生成使用 Waxman 模式，节点按照随机方式分配到系统中。本文的实验只考虑稳定状态下的系统性能。本文所采用的资源信息来源于本实验室自主研制的 Grid 搜索[13]的爬虫收集的站点资源信息，资源查找请求截取了该搜索的 2009 年 4–6 月份的用户搜索日志。表 4 是实验中所使用的参数。

表 4 模拟实验参数

参数	取值	参数	取值
d_a	4	s	1–3
QueryNum	1000–5000	m	96k
ξ	2.3	k	13
α	0.2	η	0.5

本文的目的是保证较高的成功率的同时降低查找代价、提高搜索效率，即减少系统中传递的消息数，并减少得到响应的跳数。在本文中，主要比较不同的资源查找机制在成功率、平均消息数、平均跳数三方面的性能。分别定义如下：

(1)成功率(success rate)：在整个查找过程中，被响应过的资源查找请求的比例。

(2)平均消息数(average Message Count per response, MsgC)：系统中资源查找请求每返回一个请求响应所需要的消息数的平均值。

(3)平均跳数(average Hop Count per response, HopC)：系统中资源查找请求每返回一个请求响应所经过的跳数的平均值。

4.2 路由代价分析

在 RTRSM 和 SQR 中，每一个系统中的节点均采用 **EDBF** 这个数据结构对路由的消息进行维护，两者之间的差别主要体现在维护的内容。SQR 利用 **EDBF** 对邻居共享的资源信息进行记录。在路由表建立的初期，**EDBF** 中的信息更新较为频繁。但是随着系统的不断稳定，**EDBF** 中记录的资源信息也趋于稳定。为了维持路由信息不过时，需要定期地对 **EDBF** 的内容进行更新。RTRSM 利用 **EDBF** 对邻居节点拥有的响应信息进行记录。信息更新在响应资源查找请求的同时进行。因此，RTRSM 的路由表能实时地反映了节点本身以及邻居节点感兴趣的资源，这对于提高资源查找的效率起到了重要作用。

路由表的代价包括了存储信息的代价以及维护信息的代价。由于 RTRSM 和 SQR 均通过大小一致的 **EDBF** 存储路由信息，存储信息的代价主要体现在维护响应数方面的开销。与 **EDBF** 数据结构所占用的空间相比，维护响应数的开销为常数项。因此，两者在信息存储方面的开销的量级是一致的。在维护信息方面，维护代价与信息的更新频率是成正比的。每一个更新周期 T 内，SQR 中的节点接收来自邻居节点的更新信息，因此 SQR 单位时间内的更新次数为 $n \cdot d_a / T_{\text{update}}$，在一般的路由表设置中 T_{update} 为 30 s。RTRSM 的信息更新次数与资源查找请求的频率以及查找的成功率是成正比的，可表示成：f_q.succRate.averHop，其中 f_q 是查找请求发出的频率，succRate 是查找的成功率，averHop 是查找请求平均经过的跳数。随着节点发出的请求数的增加，RTRSM 的路由信息更新次数增多，**EDBF** 中维护的路由信息更加全面，可参考性更强，从而对提高搜索效率起到了一定的作用。

4.3 查找性能分析

图 1–图 3 分析了 1000–5000 个查找请求数下 3 种不同的资源查找机制在成功率、平均消息数以及平均跳数三方面的性能。从图 1 分析可知，洪泛机制和 RTRSM 在资源查找成功率方面性能较为平稳，均分别维持在 97%和 90%以上，不随着查找请求数的增加而发生较大变化，而 SQR 在资源数较少时，成功率较高，但随着查找请求数的增加，成功率有所下降。从图 2 和图 3 分析可知，与洪泛法相比，RTRSM 大约能减少 54.2%的平均消息数，减少 56.4%的平均跳数；与 SQR 相比，RTRSM 能够减少 40.9%的平均消息数，6.4%的平均跳数。

洪泛法依赖于消息的广播，除生存时间 TTL 之外的其他因素并不能影响洪泛法所查找过的节点，因此其查找成功率、查找代价和查找效率并不随着请求数的变化而发生较大变化。SQR 通过保存节点拥有的资源信息构建提示型资源查找。在系统建立的初始状态时，各个节点已经能够预见节点周围的其他节点上的资源信息，并且该部分信息较为稳定，更新频率较低。因此，通过该部分信息能够指引转发的查找请求数相对较固定，当请求数较少时，查找成功率较高，但随着查找请求数增多，查找成功率下降。无法通过节点存储信息进行指引的查找请求，将通过广播方式进行转发，因此提高了查找代价，降低了查找效率。RTRSM 通过保存节点的请求响应信息指引请求转发，在查找请求阶段，尽量减少广播次数，减少了查询的节点数，因此与其他两种机制相比，查找成功率有所降低，但是大大降低了查找代价，提高了查找效率。通过对响应的来源进行追踪，使得无相关响应历史的查找请求能够转发给热点节点，更进一步减少了广播次数，减少了查找的代价。

图 4–图 6 分析了不同 TTL 对各种资源查找机制在查找成功率、平均消息数和平均跳数三方面的影响。从图 5，图 6 中可知，随着 TTL 发生改变，Gnutella 的洪泛法代价节节攀升，平均消息数以 11.8%的速度增加，平均跳数以 21%的速度增长；SQR 的平均消息数也发生了较大的变化，以大约

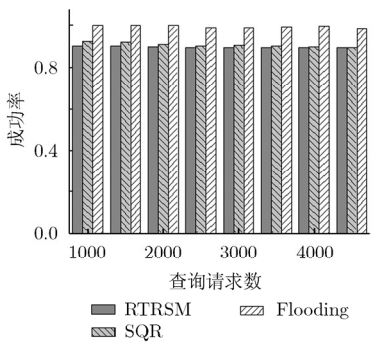

图 1 成功率

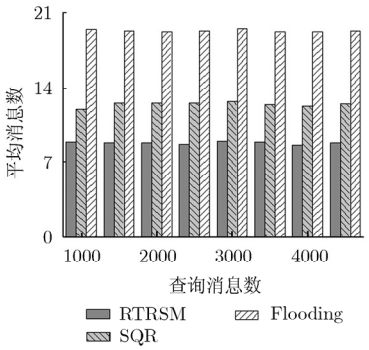

图 2 平均消息数

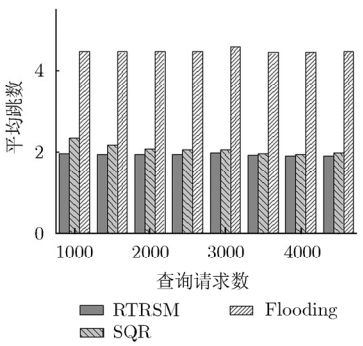

图 3 平均跳数

9.6%的速度增加，平均跳数以大约 8.9%的速度增加 RTRSM 受 TTL 影响较小，平均消息数和平均跳数均以 4.5%的速度增加。这是因为当资源查找请求转发至一个不存在相关记录的节点后，SQR 选择将此请求广播出去，而 RTRSM 则将该请求转发给热点节点，降低了广播的力度，从而降低了 TTL 对其性能的影响。

综上所述，RTRSM 通过追踪响应的响应内容以及响应的来源构建提示型资源查找，使得查找的转发更具方向性，能够保持较高的资源查找请求的成功率的同时，降低了查找的代价，提高查找的效率。同时，RTRSM 通过追踪响应的来源，识别出热点节点，更进一步地减少了广播的次数，因此 TTL 对 RTRSM 影响不大，使得 RTRSM 能够以较小的 TTL 扩展到更大规模的系统中去，而不给系统带来较大的查找代价。

5 结束语

在非结构化的 P2P 网络中，查找请求中的关键字重复出现以及资源共享不平均等现象普遍存在。本文针对非结构化的 P2P 网络中存在的这些现象提出了基于响应追踪的资源查找机制，对请求响应所响应的内容以及响应的来源进行追踪，并记录历史信息，以此构建提示型资源查找。在查找请求转发阶段，综合考虑查找请求与历史记录的相关程度以及节点的热点程度，选择最有可能返回请求响应的方向转发查找请求，在保证较高的查找成功率下，减少广播的次数，降低搜索代价，提高搜索效率。文章还对该资源查找机制进行了理论分析，分析了在该机制下，一个资源查找请求在其转发的过程中产生的消息数，并通过与实验结果进行比较而得以验证。

在 Grid 搜索中实现该基于响应追踪的资源查找机制并推广是本文的下一步工作。

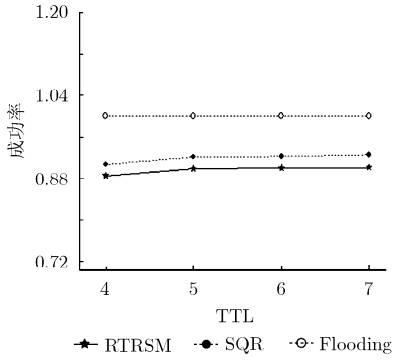

图 4 不同 TTL 时的成功率

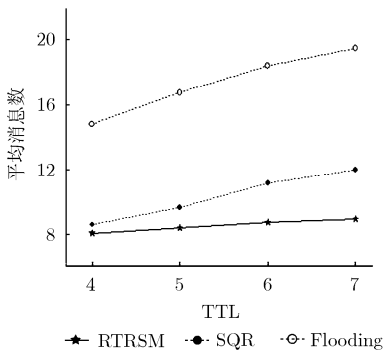

图 5 不同 TTL 时的平均消息数

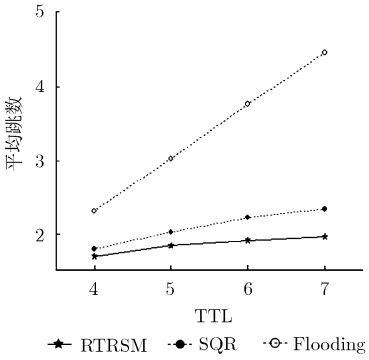

图 6 不同 TTL 时的平均跳数

参 考 文 献

[1] Clip2com. The Gnutella protocol specification v0.4. URL:http://rfc-gnutella.sourceforge.net/Development, 2010.

[2] Liu L, *et al*.. Efficient and scalable search on scale-free P2P networks. *Peer-to-Peer Networking and Application*, 2009, 2(2): 98–108.

[3] Jiang S, *et al*.. LightFlood: minimizing redundant messages and maximizing the scope of peer-to-peer search. *IEEE Transactions on Parallel and Distributed Systems*, 2008, 19(5): 601–614.

[4] Zhang Yi-ming, *et al*.. An efficient search algorithm for large-scale P2P systems. *Journal of Software*, 2008, 19(6): 1473–1480.

[5] Wang J, Gu P, and Cai H L. An advertisement-based peer-to-peer search algorithm. *Journal of Parallel and Distributed Computing*, 2009, 69(7): 638–651.

[6] Ciraci S, Korpeoglu I, and Ulusoy O. Reducing query overhead through route learning in unstructured peer-to-peer network. *Journal of Network and Computer Applications*, 2009, 32(3): 550–567.

[7] Kumar A, Xu J, and Zegura E W. Efficient and scalable query routing for unstructured peer-to-peer networks. In IEEE Infocom 2005: The Conference on Computer Communications, Los Alamitos, 2005: 1162–1173.

[8] Murat K. GnuSim: a general pupose simulator for Gnutella and unstructured P2P network. Technical Report, Department of Computer Engineering, Bilkent University, 2005.

[9] Selim C, Korpeoglu I, and Ulusoy O. Characterizing gnutella network properties for peer-to-peer network simulation. *Computer and Information Sciences*, 2005, 3733: 274–283.

[10] Yee W G, Nguyen L T, and Frieder O. A view of the data on P2P file-sharing systems. *Journal of the American Society for Information Science and Technology*, 2009, 60(10): 2132–2141.

[11] Yu Yi-jiao and Jin H. A survey on overcoming free riding in P2P networks. *Chinese Journal of Computers*, 2008, 31(1): 1–15.

[12] Brite. A network topology generator. URL: http:// www. cs. bu.edu/brite/, 2010.

[13] GSA. The Grid search. URL: http://grid.ustc.edu.cn, 2010.

王淑玲： 女，1988 年生，博士生，研究方向为 P2P 资源管理.

杨寿保： 男，1947 年生，教授，研究方向为网格与云计算、无线网络.

王喜妹： 女，1988 年生，硕士，研究方向为云计算.

武 斌： 男，1981 年生，博士，研究方向为网格资源调动.