

一种高效节能的无线传感器网络 Top-K 查询算法

程 捷 刘文予 张胜凯 蒋洪波
(华中科技大学电信系 武汉 430074)

摘 要 数据查询应用是无线传感器网络的基本应用之一,而查询最大(或最小的)的 K 个值即 Top-K 查询则是查询应用中的重要内容。高效节能是无线传感器网络应用中的关键问题。提出一种高效节能的 Top-K 查询算法 ETQA (Energy-efficient Top-K Query Approach),该方法通过数据滤波的数据融合方法预防冗余数据的上传,达到节省节点通信量的目的。系统采用数据流模型在每个时刻更新 Top-K 数据,以满足互联网上大量用户查询的需要。为了保证能量有效性,基站将适时调整节点的滤波器,阻止不必要的数据上传。同时,如果滤波器设置过高(或过低),将探寻必要的的数据以保证 Top-K 值的正确性。实际采样数据的仿真实验结果表明,该方法较 NAIVE, FILA 和 TAG 算法能够大幅度节省无线传感器网络的整体能耗。

关键词 无线传感器网络, Top-K 查询, 数据融合
中图分类号 TP393 **文献标识码** A

Energy-efficient Top-K Query Approach in Wireless Sensor Networks

CHENG Jie LIU Wen-yu ZHANG Sheng-kai JIANG Hong-bo

(Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract Data query is a basic application field in wireless sensor networks, and the greatest (or least) K value query i. e. Top-K query is one of the most important scenario in query applications. As energy-efficiency is the key problem in wireless sensor networks applications, this paper presented an energy-efficient Top-K Query approach ETQA in Top-K query. This method prevents data messages forwarding tied to message packets saving based on data aggregation with data filtering. Proposed system employs data stream model to support nearly real-time Top-K query. To save energy consumption the sink node modified filter of every node in good time. To ensure the query correctness the sink probes data from nodes when filter is set up too higher (or too lower). The performance of the proposed ETQA approach was evaluated using real data traces. The result shows that ETQA substantially outperforms the existing NAIVE, FILA and TAG in integral energy consumption.

Keywords Wireless sensor networks, Top-K query, Data aggregation

近年来,无线传感器网络得到研究者越来越多的重视,在环境污染监测、动物保护和森林防火等各种环境中的应用研究持续开展起来^[1]。Top-K 查询则是传感器网络查询应用的一个重要内容^[2]。通过 Top-K 查询无线传感器网络监测区域中最大(或最小)的多个监测量(污染物指数、声音强度或温度)可以起到判断环境中的污染程度最高地区、动物活动频繁地域或森林温度预警的作用。由于无线传感器网络节点往往布置在无人值守的环境下,降低网络整体能耗就成为需要考虑的最关键的问题之一。而通信能耗占无线传感器网络整体能耗的绝大部分,所以无线传感器网络应用的一个最关键的问题就是如何降低整体通信量。为此本文提出一种高效的节能算法 ETQA (Energy-efficient Top-K Query Approach),以降低无线传感器网络 Top-K 查询的整体能耗。

1 传感器网络 Top-K 查询

为了减少节点通信量, TAG 算法^[3]提出基本数据融合方

法。TAG 算法将某个应用(如计算所有节点采样值的均值)想要提取的信息分解为每个节点需要提取的信息(每个节点只需统计自己所有孩子节点采样值的和)来降低应用的通信量。在对无线传感器网络 Top-K 数据的监测研究^[4]中,研究者提出以下应用场景:在有 N 个节点 $d_n, n=1, 2, \dots, N$ 的传感器网络中对每个时刻 t 采集到的值 $v_n(t), n=1, 2, \dots, N$ 中查询最大(或最小)的 K 个值 $k, k=1, 2, \dots, K$ 。针对该应用提出的 FILA 算法^[4]采用了阈值的方法,即通过对每个节点设置阈值来减少无用数据的上传,节点将超过阈值的采样值数据包直接路由到基站(即 Sink 节点)。

但是,首先由于传感器网络节点的能量十分有限,要求对每个时间点 t 上的 N 个数据进行 Top-K 查询会极大增加通信量。在实际应用中,由于节点的存储能力有限,假设只能存储 T 个时刻的采样值,往往并不需要知道每个时刻 t 的 Top-K 值,而只要求查询在一个时间范围 $t' - T \leq t \leq t'$ (t' 是大于 T 的正整数)内节点采集数据中最大(或最小)的 K 个值。其

到稿日期:2009-12-11 返修日期:2010-02-26 本文受 863 国家重点基金项目(2007AA01Z223),国家自然科学基金(60803115)资助。

程 捷(1978-),男,博士,主要研究方向为无线传感器网络等,E-mail: jiecheng2009@gmail.com; 刘文予(1963-),男,教授,主要研究方向为图像处理等;张胜凯(1986-),男,硕士生,主要研究方向为传感器网络等;蒋洪波(1977-),男,副教授,主要研究方向为无线网络等。

次,互联网上多个有不同查询要求的用户希望从互联网上通过 Sink 节点查询得到结果。为此,本文采用如图 1 所示的 Top-K 查询系统,系统中的节点通过 ETQA 算法上传 Top-K 数据供不同需求的多个用户查询。而以前的算法如 TAG^[3]和 FILA^[4]难以适合以上的情景。本文第 4 节中的仿真结果将进一步说明这一点。

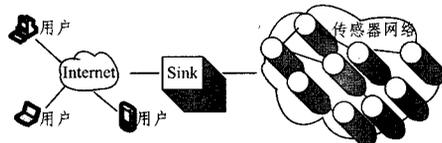


图 1 Top-K 查询系统的框架

2 Top-K 查询系统的框架

2.1 系统描述

针对本文提出的应用背景,采用图 1 所示的 Top-K 查询系统的框架。在互联网上有很多用户希望通过传感器网络的基站(或称为 Sink 节点,本文以下简称 Sink 节点)查询某个传感器网络的 Top-K 数据。为了满足用户的查询需求,Sink 节点在收到用户的查询需求后,向传感器网络发出查询要求,传感器网络会按照查询要求反馈 Top-K 数据给 Sink 节点,最后 Sink 节点将按照用户查询需求将查询得到的数据反馈给用户。本文假设网络传感器节点已建立基础的路由结构(如树状的路由结构),每个节点有固定的父节点将数据路由到 Sink 节点,不考虑无线网络的丢包和节点的移动性。在不影响一般性的情况下,假设采集到的数据均为正实数,在 $t, t'-T \leq t' < t, t' > T$ 时刻需要查询在数据 $v_n(t), n=1, 2, \dots, N, t=T+1, T+2, \dots$ 中最大的 K_t 个值。

2.2 数据流模型

由于查询用户需求的 K 值可能各不相同,假设在时间 t 同时有 Q_t 个查询需求 $K_t(q), q=1, 2, \dots, Q_t$,而每个用户可能希望在一段连续的时间点 t_1, t_2, t_3, \dots 都获得查询结果。为了满足多个查询者的要求,可以有两种实现模型。

第一种是交互模型。如图 2(a)所示,假设传感器网络由 A, B, C 3 个节点组成, Sink 节点在每次网络查询时都需要将查询要求传送到每个节点,然后节点按照要求将查询结果传送到 Sink 节点。

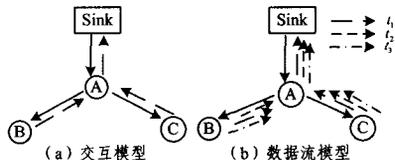


图 2 Top-K 查询的两种方式

第二种方式是数据流模型。如图 2(b)所示,在 Sink 节点一次传送查询要求后,传感器网络节点将在连续的 t_1, t_2, t_3, \dots 时刻不断向 Sink 节点报告查询结果。

显然,对于互联网上庞大的查询量 Q_t 来说,数据流模型更节省能量,产生较少网络时延和减少丢包量。本文 3.2 节将详述 ETQA 算法中 Sink 如何正确设置网络查询要求。

2.3 Top-K 数据融合

对于节点采样数据 Top-K 查询,可以借助数据融合方法,整个传感器网络在某个时间段的 Top-K 数据可以由每个节点的 Top-K 数据融合得到,如图 3 所示。

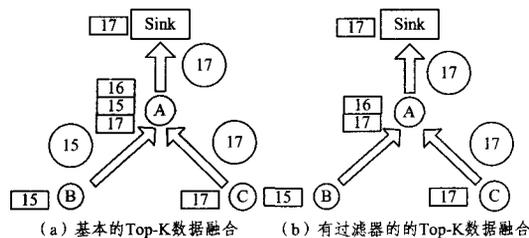


图 3 Top-K 数据融合

图 3 为传感器网络节点的数据融合示意图。假设传感器网络由 A, B, C 3 个节点组成,需要查询最大的一个数据值(即 Top-1 值)。方框中的数值表示节点采集到的数据,指向父节点的箭头附近圆形中的数据表示节点上传给父节点的数据。

在类似 TAG 的基本数据融合算法中,如图 3(a)所示, B 节点需要将最大值 15 传递给 A 节点, C 节点也将最大值 17 传递给 A 节点,在 A 节点收到 B, C 传递给自己的两个数据 15 和 17 后, A 节点将收到的数据和自己采集的数据 16 按照采样值的大小进行排序后得到最大值 17,将 17 传递给 Sink 节点即得到整个网络的 Top-K 值。

但是,如果首先给每个节点设置滤波器值 16,只有大于 16 的值才允许上传,那么 B 节点就不需要上传 15 给 A 节点,这就比简单的数据融合算法节省部分通信量,如图 3(b)所示。使用图 3(b)所示的数据融合方法,将比基本的数据融合算法节省通信量。

本文将在 3.2 节叙述 ETQA 算法中节点如何按照数据融合和在合理滤波器设置下上传数值,以及如何正确设置每个节点的滤波器值。

2.4 探寻

由于对节点设置了滤波器,而滤波器的设置有可能导致 Sink 节点得不到足够的 Top-K 值,此时需要 Sink 节点向传感器节点探寻,如图 4 所示。

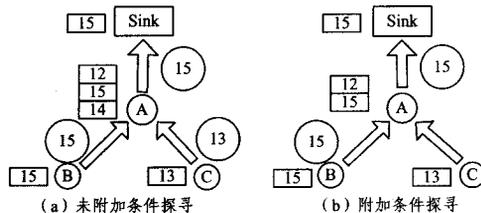


图 4 探寻

假设 A, B, C 节点的滤波器值均为 16,那么 Sink 节点会收不到数据(A, B, C 节点值均小于 16), Sink 节点必须探寻数据。若仅仅告知节点需要探寻,如图 4(a)所示,则节点 B, C 会分别上传值 15 和 13 给节点 A,此时需要两个消息包。而如果 Sink 节点在探寻时告知 B, C 节点,只需要上传大于 14 的值,那么只有 B 节点上传数据 15 给 A 节点,此时就可以节省部分通信量,如图 4(b)所示。本文将在 3.2 节详述 ETQA 算法中的探寻过程。

3 ETQA 算法

本文提出的 ETQA 算法分为两个连续的阶段,即初始化阶段和工作阶段。在初始化阶段,时间范围是 $t=1, 2, \dots, T$,通过网络所有节点向 Sink 节点传送所有的采样数据,获得工作阶段所需要的各种参数。在时刻 T 设置初始的查询 K_T

值,计算 Top-K 数据的上界 B_T ,并向各个节点传送查询条件 K_T 和各个节点的滤波器值 $f_n(T), n=1,2,\dots,N$ 。

在工作阶段,在时间范围 $t=T+1, T+2, \dots$ 的每个 t 时刻首先由 Sink 节点向整个网络传送该时刻需要查询的 K_t 值。然后从叶节点开始,网络各节点向各自的父节点传送各自未被滤波器过滤的 Top-K 值。Sink 节点由此得到该次传送的数据集 V_t 。根据此数据集, Sink 节点可能需要向网络启动探寻过程。当获得足够的数据后, Sink 节点将获得的数据存储,供用户查询。最后 Sink 节点计算整个网络的 Top-K 上界 B_t ,并更新各个节点的滤波器 $f_n(t)$,从而完成一个时刻的 Top-K 查询过程。在随后每个时刻点, Sink 节点都将收集到最新 Top-K 数据,供用户查询。

3.1 初始化阶段

在初始化过程中,节点需在时刻 $t=1,2,\dots,T$ 收集 $T \times N$ 个所有的采样数据。在每个 t 时刻,每个节点将自己采集的每个数据都上传到 Sink 节点。

在时刻 T , Sink 节点需根据数据值 Top-K 得到数据的上界 B_T ,记 Sink 此时得到的数据为 s_i ,其中数据排序为 $s_1 \leq s_2 \leq \dots \leq s_{N \times T}$ 。假设初始的查询 K 值为 K_T ,且 $K_T < N \times T$,将 B_T 设置为最大的 K_T 个数据和其他数据的分割点,也是对下个时节点采样值能够成为 Top-K 数据(如果 K_t 不变)的估计,即

$$B_T = \frac{s_{K_T} + s_{K_T+1}}{2}$$

将各个节点的滤波器值统一设置为

$$f_n(T) = B_T, n=1,2,\dots,N$$

3.2 工作阶段

工作阶段可分为 4 个过程。这 4 个过程按照顺序完成一个时刻的 Top-K 查询,然后 ETQA 算法循环 4 个过程支持连续时间点的查询工作。

1) 设置查询条件

Sink 节点首先要向各节点传送目前需要查询的 K_t 值,根据用户的 Q 个查询要求 $K_t(q), q=1,2,\dots,Q$,取最大查询值作为本次查询的 K_t 值,即

$$K_t = \max\{K_t(q), q=1,2,\dots,Q\}$$

Sink 节点将新的查询值 K_t 传送到网络的各个节点。由于每个用户的查询往往需要监视一段较长时间的 Top-K 值,而且查询要求 $K_t(q)$ 通常不会超过整个网络可查询数据总量 $N \times T$ 的 5%,所以如果查询条件 K_t 值设置足够大,例如 $N \times T$ 的 5% 就足以应对查询需求。只有当某个查询值 $K_t(q)$ 超过原有的查询值时才需要调整 K_t 。

2) 节点上传 Top-K 值

首先,叶节点 n 将采集的数据中不在时间范围 $t' - T \leq t \leq t', t' > T$ 内的数据删除,将打上已上传标记的数据去除,并将剩下的数据排序,得到最大的 K_t 个值。然后与自己目前的滤波器 $f_n(t-1)$ 对比,只将大于 $f_n(t-1)$ 的数据向父节点传送,并对向父节点传输的数据打上已上传标记。父节点在收到所有子节点上传的数据后,将收到的子节点数据合并到自己的采集的数据中排序,采用相似的步骤得到最大的 K_t 个值。并只向自己的父节点传送大于自己滤波器的值,直到最终将查询数据传递给 Sink 节点。

3) 探寻

假设 Sink 节点目前拥有的数据集 V_t 共有 M 个数据,大小顺序为 $v_t(1) \leq v_t(2) \leq \dots \leq v_t(M)$ 。如果大于 B_{t-1} 的数据

大于或者等于 K_t 个,则直接存储最大的 K_t 个数值,供用户查询,随后直接转入 4) 中所述设置滤波器过程。

如果 V_t 中大于 B_{t-1} 的数据不足 K_t 个,则说明由于滤波器的设置较高,导致小于 B_{t-1} 的数据未得到传递,而此时就需要继续探寻小于 B_{t-1} 的数据。传感器节点在收到 Sink 节点的探寻条件值后,将没有上传过(即未打上已上传标记的)并且比探寻条件值大的数据上传给 Sink 节点。将上传的节点打上已上传标记,直到 Sink 节点获得大于某个探寻值的 K_t 个最大数值,最后将该结果存储供用户查询,随后转入设置滤波器过程。

如果大于 B_{t-1} 的数据个数不足 K_t ,但拥有的数据量 M 大于或等于 K_t ,则可设置 $v_t(K_t)$ 为探寻的条件值 p_t 。按照这样的探寻条件将保证 Sink 节点在收到探寻数据后都将拥有大于 $v_t(K_t)$ 的至少 K_t 个数据。而如果大于 B_{t-1} 的数据个数不足 K_t ,而且目前拥有的数据量 M 也小于 K_t ,则需估计一个数值 $p_t = B_{t-1} \times c$ (这里 c 为一个确定 0 和 1 之间的常数,如 0.95)作为探寻条件。而此探寻条件下 Sink 节点在收到探寻结果后拥有的数据可能仍不足 K_t 个,此时则需要再次探寻,探寻条件可设为上次探寻条件的 c 倍,即 $p_t' = p_t \times c$ 。直到 Sink 拥有比某个探寻条件值 p_t' 大的至少 K_t 个数据,并且可预见的探寻次数是很有限的。

4) 设置滤波器

经过探寻过程,假设 Sink 节点此时拥有 M' 个数据 $v_t(1) \leq v_t(2) \leq \dots \leq v_t(M')$,通过 3) 说明的探寻过程可以保证 M' 大于或等于 K_t 。此时可设置新的 B_t 作为下次查询是否能成为有用 Top-K 数据的预测值

$$B_t = \begin{cases} \frac{v_t(m) + v_t(m+1)}{2}, & m=K_t, M_t' > K_t \\ v_t(m) \times c, & m=K_t, c \text{ 为常数} \end{cases}$$

为了节省通信量, Sink 节点应该只在必要的时候更新滤波器。设置滤波器可以抽象为以下的形式算法 1。

算法 1 更新滤波器

输入:上一时刻各节点滤波器值 $f_n(t-1)$,本时刻 Sink 节点收到的节点 n 上传数据集为 $X_n(t)$ 。输出:本时刻需更新滤波器值的节点 ID 集合 U_t 。

```

1: 设置  $U_t$  为空集
2: if  $B_t$  大于  $B_{t-1}$ 
3:   for 每个节点  $n$ 
4:     if  $X_n(t)$  中有大于  $f_n(t-1)$  的数值
5:       将  $n$  的 ID 加入  $U_t$ 
6:     end if
7:   end for
8: else
9:   for 每个节点  $n$ 
10:    if  $f_n(t-1)$  大于  $B_t$ 
11:      将  $n$  的 ID 加入  $U_t$ 
12:    end if
13:  end for
14: end if

```

将 U_t 中需要更新的每个节点的 $f_n(t)$ 更新为 B_t ,并将对应的滤波器值回传到相应的传感器节点。

4 实验与分析

利用 Matlab 仿真来验证算法的有效性。工具仿真放置在英特尔伯克利实验室采集到的 2004 年 3 月 1 日一天所采集到的温度数据^[5]作为实验数据,该数据使用 54 个 Mi-

ca2Dot 型传感器节点,每个节点每隔 31s(一个时间点)采集一个温度数据,传感器节点接受数据包的能耗为发送数据包能耗的 0.37 倍^[6]。设置传感器保持 120 个最近时间的数据,查询系统需要查询 K 个最大的数据,并假设在查询过程中 K 的固定值不变,评估在 Top-K 查询的过程中所有传感器节点传送和收到数据包的整体能耗。

传感器节点上传到父节点的数据包包括一个 32 比特的数据值部分和一个 32 比特的数据 ID 部分,其中数据 ID 部分包括一个 16 比特的节点 ID 和一个 16 比特的时间戳。Sink 节点回传到传感器节点的探寻包和更新滤波器包均为一个 32 比特的数据值(探寻条件值或滤波器值)。由于传感器节点收发数据包的能耗与数据包的大小成正比,为了简便起见,将发送一个 32 比特包定义为一个通信包,将收到一个 32 比特包折算为一个通信量的 0.37 倍,通过比较通信量可以评估算法的整体通信能耗。

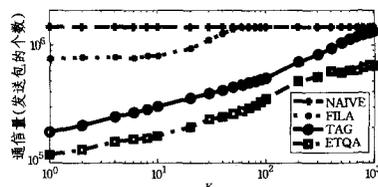


图 5 算法通信量仿真结果比较

为了与 ETQA 算法对比,实验同时仿真了 NAIVE、FILA 与 TAG 3 种算法作为对比。NAIVE 算法即无线传感器网络中所有节点将采集到的所有数据都上传到 Sink 节点的方法,简称 NAIVE 算法,横轴为 1 到 1000 之间选择的 28 个 K 值,纵轴为 2004 年 3 月 1 日一天数据的 Top-K 查询中所有节点的整体通信量,如图 5 所示。

(上接第 88 页)

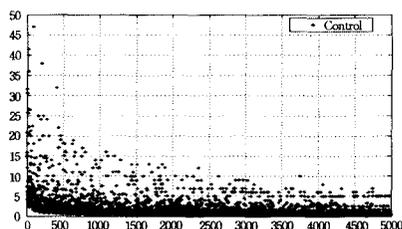


图 4 控制算法下结点负载分布情况

结束语 本文首先指出针对高异构性的 P2P 网络,连接数根本不能较准确地反映网络中的“集散节点”,所以我们采用节点的负载度来进行衡量,并将负载度高于邻居负载均值的节点称为重载节点。采用均衡控制算法将它的一部分负载向轻载节点转移。相比未采取控制算法的 P2P 网络,节点负载度分布十分均衡。我们下一步的工作首先要考察节点能力值的时刻变化和高动态性情况下算法的运行情况,还要尝试把此均衡控制算法应用于结构化 P2P 网络。

参考文献

- [1] Adar E, Huberman B. Free riding on Gnutella[J]. First Monday, 2000, 5(10): 32-35
- [2] Ramaswamy L, Liu L. Free riding: A new challenge to Peer-to-Peer file sharing systems[C]//Proceedings of the 36th Hawaii International Conference on System Sciences, Hawaii, 2003; 220-229
- [3] Li Zhen-hua, Chen Gui-hai, Qiu Tong-qing. Partition Nodes: Topologically-critical Nodes of Unstructured Peer-to-Peer Net-

通过通信量对比仿真发现,本文提出的 Top-K 查询算法 ETQA 比 NAIVE、TAG 和 FILA 算法能较大幅度地减少网络节点的整体通信量,从而大大地节省了传感器节点的整体能耗。由于 ETQA 算法中只对数据排序和比较,数据处理延时很小,并且算法中探寻和设置滤波器次数非常有限,同时通信数据量较小(如仿真中 31s 才采样一个数据),通信延时也较小,因此整体上 ETQA 算法的时间复杂度与其他的 3 种算法相似。

结束语 为了节省通信量,本文提出一种结合具体应用的数据融合算法,以利用数据的相关性通过设置滤波器达到减少冗余数据的上传的目的。使用数据融合方法降低无线传感器节点整体通信量需要结合具体问题具体分析。

参考文献

- [1] Sakai H, Iiyama S, Tokoc K. Evaluation of Water quality and pollution using multichannel sensors[J]. Sensors and Actuators B: Chemical, 2000, 66: 1
- [2] Silberstein A, Braynard R, Ellis C, et al. A sampling-based approach to optimizing top-k queries in sensor networks[C]//Proceeding of IEEE ICDE, 2006
- [3] Madden S, Franklin M J, Hellerstein J M, et al. TAG: A Tiny Aggregation Service for Ad Hoc Sensor Networks[C]//Proc. Usenix Fifth Symp. Operating Systems Design and Implementation (OSDI '02). Dec. 2002; 131-146
- [4] Wu M, Xu J, Tang X, et al. Top-k monitoring in wireless sensor networks[J]. IEEE Trans. on Knowledge and Data Engineering, 2007, 19: 7
- [5] <http://db.csail.mit.edu/labdata/labdata.html>
- [6] http://www.xbow.com/products/Product_pdf_files/Wireless_pdf/MICA2DOT_Datasheet.pdf
- [7] Liu X, Xiao L, Kreling A, et al. Optimizing Overlay Topology by Reducing Cut Vertices[C]//Proceedings of ACM NOSSDAV, 2006
- [8] Hochman H M, Rodgers J D. Pareto Optimal Redistribution[J]. The American Economic Review, 1969, 59(4): 542-557
- [9] Yu Yi-jiao, Jin Hai. A Survey on Overcoming Free Riding in Peer-to-Peer Networks[J]. Chinese Journal of Computers, 2008, 31(1)
- [10] Liu Yuhua, Yang Chun, Xu Kaihua, et al. Model of Controlling the Hubs in P2P Networks[J]. Journal of Networks, 2009, 4(4)
- [11] Iamnitchi A, Ripeanu M, Foster I. Small-world file-sharing communities[C]//IEEE infocom, 2004
- [12] Hui Y K, Lui C S, Yau K Y. Small-world overlay p2p networks [R]. CS-TR-2004-04. The Chinese University of Hong Kong, April 2004
- [13] Kempe D, Dobra A, Gehrke J. Gossip-Based Computation of Aggregation Information[C]//Proceedings of IEEE FOCS, 2003
- [14] 陈林书,柳媛慧. P2P 网络中基于节点能力自适应的搜索算法[J]. 湖南科技大学学报:自然科学版, 24(2)
- [15] 任超,李战怀,张英. 异构 P2P 网络的分布式查询协议[J]. 电子科技大学学报, 38(1)
- [16] Chawathe Y, Ratnasamy S, Breslau L, et al. Making gnutella-like P2P systems scalable[C]//Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Karlsruhe, Germany, August 2003; 25-29