

机会传感器网络中基于优先级的查询分发和处理算法

赖永炫¹, 林子雨²

(1. 厦门大学 软件工程系, 福建 厦门 361005; 2. 厦门大学 计算机系, 福建 厦门 361005)

摘要: 提出了机会无线传感器网络中一种基于查询优先级的查询分发和处理(PBQ)算法。算法根据用户定义的查询优先级合理选取消息转发的中间节点, 并控制查询消息的转发和拷贝量; 同时, 动态地调整查询消息的优先级, 使得查询结果能快速回传到查询发起者并清理网络中查询的残留消息, 避免了不必要的消息传输并提高查询处理的整体效用。模拟实验表明 PBQ 算法能有效提高机会网络中查询处理的成功率, 降低查询的开销和时间延迟。

关键词: 查询优先级; 查询处理; 机会网络; 传感器网络

中图分类号: TN929.52

文献标识码: A

文章编号: 1000-436X(2014)03-0085-09

Priority-based message forwarding and query processing in opportunistic wireless sensor network

LAI Yong-xuan¹, LIN Zi-yu²

(1. Department of Software Engineering, Xiamen University, Xiamen 361005, China;

2. Department of Computer Science, Xiamen University, Xiamen 361005, China)

Abstract: A priority-based algorithm called PBQ was proposed for the query forwarding and processing in opportunistic wireless sensor network. Through the user-defined query priorities, the proposed algorithm properly selects the relay nodes and controls the forwarding and the amount of messages. Meanwhile, the query priorities were adjusted dynamically so that the query results could be forwarded back to the source node of the query quickly and the residual query request messages could be cleaned up from the network, saving lots of unnecessary transmissions and improving the overall utility of query processing. Simulating results show that PBQ could effectively improve the success rate of queries, and reduce the cost and delay of query processing in opportunistic wireless sensor network.

Key words: query priority; query processing; opportunistic network; wireless sensor network

1 引言

机会无线传感器网络(opportunistic wireless sensor network)是节点带有感知功能、主动或被动的选择机会主义通信方式进行数据传输的一种机会网络。比如, 在移动传感器网络(mobile wireless sensor network)中, 节点的移动、信号干扰等因素容易导致间断性的通信链路; 而基于节能的考虑, 很多应用也会主动关闭节点的无线射频装置, 导致网络的

断开^[1]。机会网络中不存在端到端的通路, 需要采用机会主义的通信方式进行数据和消息的传输, 因此它更加适合自组织网络实际部署的需求。典型的应用有: 车载网络^[2]、野生动物追踪^[3]、农地数据收集^[4]、偏远地区网络传输^[5]等。

传感器网络是以“数据为中心”的网络^[6]。节点会不断地产生感知数据, 针对传感器网络的查询和处理是一个关键性的问题。在静态传感器网络中, 典型的查询处理算法如 TAG^[6]、PULL/PUSH^[7]

收稿日期: 2012-10-30; 修回日期: 2013-09-30

基金项目: 国家自然科学基金资助项目(61202012, 61303004); 福建省自然科学基金资助项目(2011J05156, 2013J05099); 中央高校基本科研业务费专项基金资助项目(2012121030)

Foundation Items: The National Natural Science Foundation of China(61202012, 61303004); The Natural Science Foundation of Fujian Province (2011J05156, 2013J05099); Fundamental Research Funds for the Central Universities(2012121030)

等都依赖于路由树、聚簇等网络基础设施来进行查询的分发和查询结果的收集；而在机会网络中，不存在稳定的通信链路，难以建立和维护类似的支持查询处理的基础设施。因此传感器网络中已有的查询处理算法无法适用于机会传感器网络中。这给机会无线传感器网络中的查询处理带来了很多挑战性的问题，主要体现在以下几方面。

1) 每个节点都可以是查询的发起者和回答者，查询消息没有预订的目标地址。只能通过类似洪泛的方式进行数据传播，这常导致过量的消息复制和传输。

2) 网络中存在多个查询，而有限的网络传输机会将导致一些查询无法被处理。如何协调查询间的差异以最大化查询处理的综合效用，是一个挑战性的问题。

3) 机会网络依赖于“存储-携带-转发”的转发机制。节点的缓存有限，一些数据和查询结果可能因为缓存空间不足丢弃，导致查询结果的丢失。

4) 已有的工作大都集中在机会网络中通用的消息分发和转发机制，专门针对机会无线传感器网络查询处理的研究还比较少。

在查询处理应用中，查询具有相应的语义信息，且查询结果需返回给查询的请求者。已有的研究不区分查询的优先级顺序，常导致重要查询得不到及时的处理，降低网络查询处理的整体效用。因此，给查询赋予不同的优先级级别，结合查询处理本身的特点来优化查询过程中的消息分发和查询执行，能够降低查询处理的成本和时间延迟。本文研究了机会无线传感器网络中的查询处理问题，并提出了一种基于查询优先级的查询分发和处理(PBQ, priority-based query forwarding and processing)算法。PBQ 根据用户自定义的查询限定时间和重要性系数定义查询的优先级，并据此合理选取消息转发的中间节点，控制查询消息的转发和拷贝量以优化查询的分发；同时，算法也动态地调整查询消息的优先级，使得查询结果能快速回传给查询发起者并主动清理网络中的查询残留消息，以避免不必要的消息传输，提高查询处理的整体效用。模拟实验表明 PBQ 算法能有效提高机会网络中查询处理的成功率，降低查询开销和时间延迟。

2 相关工作

机会网络大都采用“存储-携带-转发”的模式

进行消息的转发。在该模式中节点可将消息缓存在当前节点上，并在合适的时刻转发给其他节点，直至消息成功转发到目的节点^[8-10]。传染性路由协议^[11]采用类似洪泛的机制，相遇的节点充分利用节点接触时的通信机会来彼此交换所携带的消息。为避免洪泛带来的链路拥塞，受控洪泛(controlled flooding)的算法^[12]则利用转发概率、生存时间和死亡时间等参数选择性地转发消息；同时当消息到达目的节点时，利用 Cure-Ack 回复阻止其他节点继续转发该消息。PREP(priority-based epidemic)协议^[13]是一种基于优先级的传染性路由协议，它根据到达目的节点的成本、已离开源节点的成本、过期时间等对消息块进行优先级排序，并按优先级进行消息的转发和删除，使得靠近目的节点时拷贝的密度增加。

另一类数据转发算法则尽可能地利用网络的知识 and 上下文环境信息，使消息转发到与目标节点相遇概率高的节点。ZebraNet^[3]提出了基于移动历史的转发机制，每个动物身上的传感器节点维护一个到达 sink 节点的概率，节点相遇时概率低的节点将消息转发给概率高的传感器节点。PROPHET 算法^[14]通过观察节点间接触的历史来计算传输的预测概率，仅当邻居节点的预测概率高于自身时，当前节点才把消息传输给该邻居节点。文献[15]提出了一种基于社区的消息传输算法 CMTS，能够根据节点之间的通信频繁程度，自动将节点划分成不同的社区，自适应地控制消息的拷贝数量并依靠活跃节点将消息传输到目标社区。此外，也有基于社会网络分析方法的消息传输算法^[16-18]。此类算法利用中心度(centrality)作为基础指标，定义各节点的活跃程度并选择合适的中转节点。比如，BUBBLE Rap^[18]协议以中介度(betweenness)作为主要衡量指标；而基于友谊的路由协议^[16]则根据行为定义节点间的友谊关系，以指导节点间的消息转发和通信。

然而，以上介绍的机制都是针对网络中通用的消息传递和转发，并未对查询处理的消息分发进行专门的优化。机会网络中查询处理方面的研究相对较少，且大多使用基于内容的路由方式传输数据。文献[19]引入信息检索领域的语言模型，算法将检索请求表示成关键字集合，文档则表示成一个随机过程。根据文档与多个查询请求的匹配程度计算文档优先级，并作为缓存管理和传输调度的依据。DelQue 算法^[20]采取一种两跳回环的方式来回答查询：基于社会网络分析的方法利用 semi-Markov 过程预测并

选择代理节点，由它负责查询分发，并把相关查询结果返回给发起者。文献[21]提出了机会网络的订阅发布系统，算法利用社会网络分析的方法选举代理节点，并由它们负责订阅信息的收集和分发；通过动态控制代理节点的数量，算法可以有效权衡订阅系统的效率和代价。然而，以上算法并未考虑查询本身的优先级问题及未采取相应的优化措施。

3 查询的优先级模型

机会无线传感器网络中的查询，由查询发起者 (requester) 发出查询请求 re ，并以机会主义式的通信方式传播该请求；收到请求的节点查询本地数据，如果回复节点(replier)拥有回答 re 的结果 rs ，则通过机会主义的通信方式把 rs 返回给查询发起者。网络存在查询请求 re 和查询结果 RS 的一个映射： $h(re)=RS$ 。为了简化起见，假设每个回答节点都拥有回答查询的全部数据，即 $rs=RS$ 。

以时间和重要性为象限，可粗略地把查询分成以下几类（如图 1 所示）。

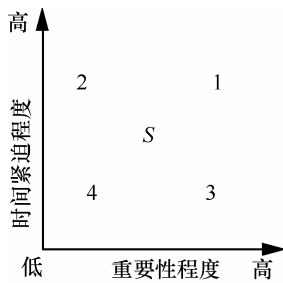


图 1 查询的分类示意

- 1) 时间紧迫，重要性程度高。
- 2) 时间紧迫，但重要性程度不高。
- 3) 时间不紧迫，但重要性程度高。
- 4) 时间不紧迫，重要性程度也不高。

每一个查询类别，其所需要的查询资源也应该不同。为了对查询的优先级进行形式化的定义，首先定义一个标准查询(standard query)。

当查询 $S(re, rs, st, sw)$ 定义为默认的重要性程度 sw ，消耗了一定的查询资源，在默认时间 st 内能将查询请求 re 发出，并返回查询结果 rs ，称查询 S 为标准查询。

图 1 中的 S 位置示意了标准查询在“时间紧迫程度”和“查询重要性程度”象限中的位置。基于标准查询 $S(re, rs, st, sw)$ ，查询 $Q(re, rs, t, w)$ 可以按式(1)定义查询的优先级

$$P(Q(t, w)) = \alpha \frac{st}{t} + (1-\alpha) \frac{w}{sw}, t \in [st/T_1, \infty), w \in [0, T_2 \cdot sw] \quad (1)$$

其中， α 是自定义的权重系数， t 为查询的限定时间， w 为查询重要性程度， T_1 和 T_2 是 $\frac{st}{t}$ 和 $\frac{w}{sw}$ 所能取值的最大值。可以看出，查询优先级 P 与 t 逆相关，与 w 正相关。标准查询的优先级 $P(S(st, sw))=1$ ；而一般的查询，其满足以下条件

$$0 \leq P(Q(t, w)) \leq \alpha T_1 + (1-\alpha) T_2 \quad (2)$$

标准查询的重要性程度 sw 由用户指定，定为 1.0，并以此来定义其他查询的重要性程度；标准查询的时间 st ，可取若干成功查询所用时间的平均值来表示。在此，假设一般查询的优先级系数 t 和 w 是用户预先定义的，用户可根据查询请求 re 和查询结果 rs 的内容，根据语义规则 f 计算查询的限定时间 t 和重要性程度 w ，即 $f(re, rs) \rightarrow (t, w)$ 。本文主要研究的是查询消息的分发算法，限于篇幅限制， $f(re, rs) \rightarrow (t, w)$ 的映射问题留在以后的工作中讨论。

4 基于优先级的查询处理算法

4.1 算法概述

机会网络由于不存在稳定的端到端的稳定路由，采用“存储—携带—转发”的策略进行机会主义式的消息传递，查询处理会产生一定的延迟，甚至无法完全保证查询的成功。鉴于此，本文提出的查询处理 PBQ 算法在各节点采用基于优先级的策略进行消息的转发和查询处理。算法的基本路由和处理策略是：根据查询的优先级，控制消息的分发和拷贝量；并根据节点的活跃程度动态更新消息的优先级，合理选取消息分发的中间节点；在查询成功时清理网络中的查询残留消息，从而优化查询处理的时间，提高查询处理的成功率和整体效用。具体来讲，查询处理可分解为以下 8 个步骤（如图 2 所示）。

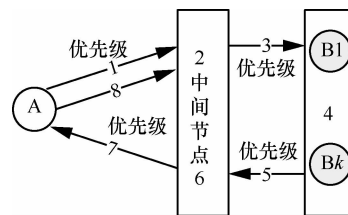


图 2 查询处理示意

1) 查询请求者 A, 根据节点的活跃程度分配查询消息的优先级, 并把查询消息分发给中间节点。

2) 中间节点收到查询, 并根据相应的查询优先级进行转发。

3) 中间节点根据查询消息的优先级, 把查询请求分发到能回答查询的节点 B。

4) 节点 B 在本地进行查询处理, 并获取查询结果。

5) 节点 B 定义查询结果消息的优先级, 并把结果消息转发给中间节点。

6) 中间节点收到查询结果, 根据节点的活跃程度分配结果消息的优先级, 并进行消息转发。

7) 中间节点把查询结果发送给查询发起者, 该查询处理成功执行。

8) 节点 A 发出查询清理消息, 清理网络中残留的关于该查询的消息, 查询处理完毕。

以下分初始运行、查询分发、查询执行与返回、查询清理等 4 个阶段具体介绍 PBQ 算法。

4.2 初始运行

在初始阶段, PBQ 采用类似传染性路由^[11]的机制传输消息并收集与网络和节点相关的元数据信息。每个节点维护一个接触者列表 CL(contact list)。CL 按接触频繁度的高低, 记录了与该节点接触的节点 id、接触时间等信息。同时, 在接触频度的基础上, 根据节点的可用缓存空间来计算节点的活跃系数 A

$$A(s_i) = \gamma \frac{N(s_i)}{N_{\max}} + (1 - \gamma) \left(\frac{usize(s_i)}{size(s_i)} - 0.5 \right) \quad (3)$$

其中, γ 是预定义的权重系数, $N(s_i)$ 表示观察时间段内可以与节点 s_i 消息传输的节点个数, N_{\max} 是网络中单个节点可能拥有的最大传输节点数, 即 $N_{\max} = \max(N(s_k)), s_k \in S, S$ 为网络节点的集合。 $size(s_i)$ 表示节点 s_i 的缓存大小, $usize(s_i)$ 表示节点 s_i 未使用的缓存空间大小。活跃系数考虑了缓存空间的因素, 平衡了接触机会和缓存大小的因素, 有利于综合地利用节点的通信和存储资源。

此外, 算法中节点生成或接收的查询消息先存放在缓存中, 在获得通信机会时才把消息复制到转发队列中等待分发。缓存以消息为单位组织, 每条消息分组含: <查询 id, 查询信息 query, 类型 type, 内容 content>。表 1 列出了查询消息的类型。

类型	内容
RE	查询请求的消息
RS	查询结果的消息
KQ	清理查询的消息

4.3 查询分发

算法中每个查询请求都包含在 1 个消息分组 p 中。消息 p 包含查询的基本信息 $Q(re, t, w)$, 以及 <Receiver, QP, Prior> 等字段。其中, Receiver 表示查询结果的优先接收者, 即请求者的 id 和接触者列表的集合 ($src \cup src.CL$), QP 表示查询初始时的优先级 $P(Q(re, t, w))$, Prior 表示消息 p 自身的优先级。初始时, 消息的优先级 Prior 设为查询本身的优先级: $p.Prior = P(Q(re, t, w))$, 它将随着消息的传递而动态地更新。

如图 3 所示, 当节点 s_i 与另一节点 s_j 相遇并可进行通信时, 节点 s_i 与 s_j 将交换与查询相关的元数据信息, 并据此选取将要传输的消息分组。元数据信息包括查询 id, 节点的活跃系数 A 等。由于网络中的每个查询都有唯一的 id, 通过比较相遇节点中存储的查询 id 列表, 节点 s_i 可以从缓存中选取节点 s_j 所没有的查询消息分组, 对它们进行过滤并复制一份移入转发队列中; 对于节点 s_j , 也按类似的方法处理。本文以存储在 s_i 中的消息 $p1$ 为例, 分发算法按以下 4 个步骤进行。

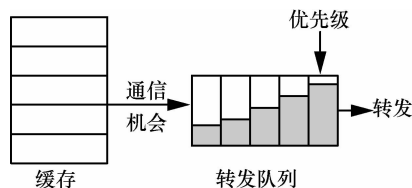


图 3 节点 s_i 的消息缓存和转发模型

1) s_i 首先复制 $p1$ 为 $p2$, 并设置 $p2$ 的优先级。如果当前能判断 s_j 可以回答 $p2$ 所包含的查询, 则 $p2$ 的优先级设置为: $p2.Prior = \text{MAX}$, MAX 表示优先级的最大值; 如果无法判断, 则 $p2$ 的优先级设置为: $p2.Prior = p.Prior \frac{A(s_j)}{A(s_i) + A(s_j)}$ 。当 $p2$ 的优先级大于某个阈值: $p2.Prior > p1.QP \cdot mp$, $p2$ 移入转发队列中, 其中, mp 是用户自定义的阈值。

2) s_i 的转发队列根据消息的优先级排序, 并按优先级的先后顺序把消息 $p2$ 从 s_i 传输到 s_j 。

3) 如果 p_2 传输成功, s_i 将更新消息 p_1 在缓存中的优先级: $p_1 \cdot \text{Prior} = p \cdot \text{Prior} \frac{A(s_j)}{A(s_i) + A(s_j)}$ 。

4) 当通信机会结束, s_i 节点将清空转发队列中未转发的消息。如果接收消息的节点 s_j 缓存空间不足时, 将根据优先级顺序淘汰优先级最低的消息。

可以看出, 查询请求消息的优先级按照节点活跃系数在传输节点之间分配。消息按优先级排序, 优先级高的消息将位于转发队列的头部, 尽早分发出去; 而优先级较低的消息, 将无法被拷贝到转发队列, 或处于转发队列的尾部而无法被转发出去。因此, 基于优先级的转发机制, PBQ 算法能自动过滤掉那些低优先级的消息并控制消息的转发数量。

4.4 查询执行和返回

收到查询请求消息的节点, 将根据查询检索本地存储的数据回答查询。为了简化起见, 假设每个节点要么不能回答查询, 要么拥有回答查询的全部数据。如果节点无法回答查询, 则查询消息 p 将按照 4.3 节的方法继续分发到其他节点; 如果节点可以回答查询, 则节点将生成查询的结果消息分组 p' , 并更新相应的消息优先级。

查询结果消息分组 p' 与查询请求的消息分组类似, 除了包含查询的基本信息 $Q(re, t, w)$ 、字段 $\langle \text{Receiver}, QP, \text{Prior} \rangle$, 还包含查询结果 rs 。 p' 的初始优先级设为查询本身的优先级: $p' \cdot \text{Prior} = P(Q(re, t, w))$, 并插入到节点的缓存中。与查询请求消息分组不同的是, 查询结果消息分组需返回到查询的请求节点。消息分组字段 Receiver 包含了请求者的 id 和接触者列表 CL 。因此, 当节点 s_i 与 s_j 相遇时, 除了交换基本查询相关的元数据信息, 节点 s_i 还将从缓存中寻找满足以下条件的消息分组 p_3

$$p_3 \cdot \text{Type} = RS; s_j \in p_3 \cdot \text{Receiver} \quad (4)$$

其中, RS 表示该消息分组含查询结果。 p_3 将被复制为 p_4 , 并暂时性地提升其优先级, 然后插入到转发队列中

$$p_4 \cdot \text{Prior} = \begin{cases} \text{MAX}, s_j = \text{src} \\ p_3 \cdot \text{Prior} \cdot QP, s_j \in \text{src} \cdot CL \end{cases} \quad (5)$$

其中, MAX 表示优先级的最大值; QP 表示查询初始时的优先级 $P(Q(re, t, w))$ 。查询结果消息分组 p_4 的分发可分为 3 类情形进行讨论。

1) 当携带查询结果消息分组 p_4 的节点 s_i 遇到查询发起者 src 时 ($s_j = \text{src}$), p_4 将以最大的优先级

传输到发起者节点, 完成查询; 同时进入查询清理阶段 (见 4.5 节)。

2) 当 s_i 遇到的节点 s_j 属于查询发起者的接触列表中的节点时 ($s_j \in \text{src} \cdot CL$), p_4 的优先级将在 p_3 的基础上临时提高 QP 倍。但当 p_4 成功传输到节点 s_j 时, 消息 p_3 和 p_4 的优先级仍按照节点间的活跃系数进行分配, 即在节点 s_i 中, $p_3 \cdot \text{Prior} = p_3 \cdot \text{Prior} \frac{A(s_j)}{A(s_i) + A(s_j)}$; 在节点 s_j 中, $p_4 \cdot \text{Prior} = p_3 \cdot \text{Prior} \frac{A(s_j)}{A(s_i) + A(s_j)}$ 。

3) 如果相遇节点 s_j 不属于以上 2 种情况, 则消息分组 p_4 分发方法与查询请求的分发方法类似。即消息的优先级按照节点活跃系数在传输节点之间分配, 并依据消息优先级决定转发的次序。同时, 查询结果 p_4 可用于清理网络中残留的查询请求消息分组。即节点 s_i 中满足以下条件的消息分组 p_5 都将被删除掉

$$p_5 \cdot \text{Type} = RE; \text{QueryId}(p_5) = \text{QueryId}(p_4) \quad (6)$$

其中, $\text{QueryId}(p)$ 表示消息 p 包含查询的 id 。

4.5 查询清理

当查询的发起节点接收到查询结果时, 该查询执行成功。但此时网络中还存在着查询的请求信息、查询结果等遗留消息, 因此还需要一个查询的清理阶段, 以避免重复的消息传输。

假设 s_i 是查询的发起者, 当它第一次收到从节点 s_j 传来的查询结果消息 p_4 时, 则会生成一个查询清理消息分组 p'' 。 p'' 的类型设置为 KQ , 表示查询清理消息; 它包含已经成功执行查询的 id , 并设置为最大的优先级: $p'' \cdot \text{Prior} = \text{MAX}$ 。 p'' 会立刻插入到转发队列中, 以回传给节点 s_j ; 同时拷贝一份存储在节点 s_i 的缓存中。当节点收到查询清理消息分组时, 将从缓存中删除与 p'' 具有相同查询的消息分组。即删除缓存中具有相同查询 id 的消息分组

$$\text{QueryId}(p) = \text{QueryId}(p'') \quad (7)$$

查询清理消息 p'' 转发时, 其优先级的分配和更新与 4.3 节查询分发的消息一致。由于它具有较高的初始优先级, 因此可以在网络中快速转发, 以清理查询的遗留消息。值得指出的是在 PBQ 算法中, 随着消息的转发, 查询请求和结果消息的优先级会因为不断的分割而降低, 从而通过缓存淘汰策略自动地被删除掉。

5 实验结果与分析

5.1 环境设置

用 C#语言进行仿真实验，以验证 PBQ 算法的性能和有效性。实验中，节点被划分为 K 个小组，每个小组拥有一块预定义的区域 (group area)。属于同组的节点，将以概率 g_i 在小组区域内采用 Way-Point 模型随机移动，以 $1-g_i$ 的概率移动到区域外移动；且每隔一段固定时间，节点将回到该小组区域的中心点。节点不断感知外界温度，但只有最近 100 次读数被保留在节点的存储中。查询定义为：给定某节点的 id ，返回该节点感知温度排名 TOP10 的数据。标准查询 S 的执行时间定义为 200 s，优先级定义为 1.0。查询的限定时间 t 和优先级 w 取值均大于 0，且服从正态分布： $t \sim N(200, 200)$ ， $w \sim N(1, 0.5)$ 。实验以 0.03 查询/秒的平均速率 (查询产生率) 随机生成查询请求节点和回答节点的 id 。

由于通信开销是传感器网络节点能量消耗的主要因素，因此本文仅以网络发送数据分组数量作为算法能量代价的衡量标准。实验中假设可通信的节点间具有理想的数据链路，不考虑传输时的出错和延迟问题；但通信时有带宽限制，即每次通信机会节点所能传递的消息数有所限制。同时，实验规定每个查询请求、查询结果以及节点元数据信息交换均需 1 个消息分组进行传输。表 2 列举模拟实验中各参数的默认设置。

表 2 实验参数设置

参数	取值	说明
N	50	网络节点个数
W	90	模拟区域: 90 m × 90 m
K	10	划分小组的个数
R/m	6	节点的通信半径
Sim_T/s	2 000	模拟时间的总长度
$\min, \max/m \cdot s^{-1}$	1, 5	节点运动速度的最小值和最大值
$bandwidth/packet \cdot s^{-1}$	[5, 50]	节点的带宽区间
g_i	0.6	节点在小组区域移动的概率
α	0.5	计算优先级的权重系数
γ	0.7	计算活跃度的权重系数
mp	0.1	用户自定义的优先级阈值
MAX	10	优先级的最大值
$BufferSize/packet$	200	缓存大小
$QueueSize/packet$	30	转发队列的大小

由于尚未发现机会无线传感器网络中基于优先级的查询算法的研究，对比算法基于现有的典型数据分发算法 (Epidemic 算法^[11]、PROPHET 算法^[14]) 进行查询处理，并采用“先进先出”的缓存更新策略。实验结果表明，与基于传染性和概率性路由的查询处理算法相比，PBQ 算法能有效提高查询的成功率，并降低查询消息的传输量和时间延迟。

5.2 实验结果分析

网络中的查询量用查询产生率表示，即每个节点平均每秒产生的查询数量。如图 4 所示，随着查询产生率的增大，各算法的消息传输量也增加。但 PBQ 算法的总传输量最小，约为 Epidemic 算法的 72%~81%。节点相遇时，Epidemic 算法会尽可能多地传输消息；而 PBQ 算法在消息分发时，则按查询的优先级进行传输，且设置了能够进行转发的优先级阈值，因此减少了消息转发的数量。在返回查询结果时，通过请求节点的相遇列表 CL，能选择性地快速回传；并且查询结果返回到请求节点时，节点会发送 KQ 消息，主动清理该查询的请求消息，以避免不必要的消息传输。

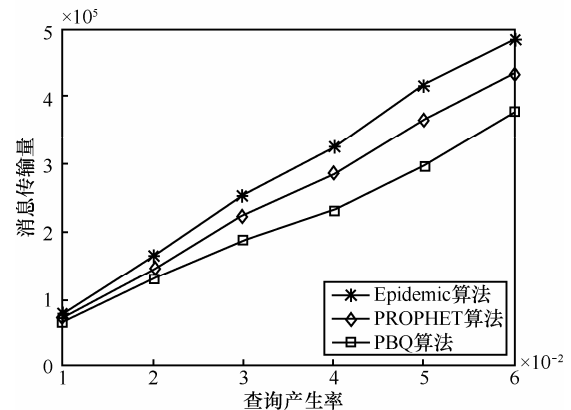


图 4 查询产生率对消息传输的影响

图 5 则比较了查询处理的成功率。可以看出，随着查询数量的增大，各算法的查询成功率降低；但 PBQ 算法的查询成功率比其他算法大约高出约 5%~18%，达到 63%左右。当网络中查询较多时 (较大的查询产生率)，PBQ 算法的优势比较明显。这一方面是基于优先级的消息传输算法和缓存淘汰策略有利于查询消息在最初阶段快速扩散，并在获得查询结果时能有效返回到请求节点。而在 Epidemic 算法和 PROPHET 算法中，大量的查询消息会因为缓存的淘汰策略而丢失，从而降低了查询成功率。

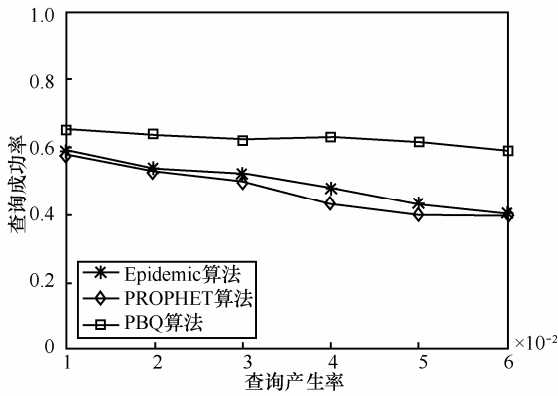


图 5 查询产生率对查询成功率的影响

实验还比较了各查询算法的平均延迟时间，即从查询发出到节点收到查询结果所用的时间。图 6 给出了各算法所有查询的平均延迟时间，可以看出，所有查询的延迟随着查询产生率的提高而增加。这是由于查询增大时，查询的成功率降低了；而未成功执行的查询，其延迟时间设为网络的总模拟时间 2 000 s。因此，由于 PBQ 算法的查询成功率最高，其查询时间延迟也最低，在 800~1 000 s 左右。对于成功执行的查询，各算法的平均延迟时间差别不显著，大都落在[220 s, 300 s]的区间内。

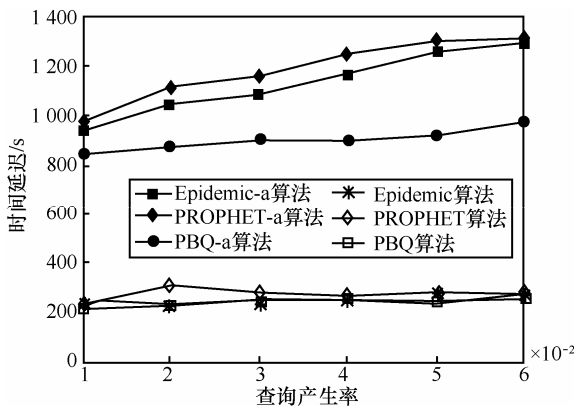


图 6 查询产生率对延迟时间的影响

由于各查询算法都是基于机会网络中“存储—携带—转发”机制进行消息分发的，本文在实验中也分析了缓存大小对各算法性能的影响。如图 7 所示，随着缓存的增加，各算法的消息传输数有所增加。PROPHET 算法采用了基于概率的传输，其传输量随缓存略有增加；在 Epidemic 算法中，节点会充分利用每个通信机会交换彼此没有的消息，因此它的传输量也增加得最多。缓存为 500 个消息分组时，Epidemic 算法的消息传输量比缓存为 100 个消息分组时约增加了 80%的传

输量。PBQ 算法的传输量也增大，但由于被传输的消息需满足一定的优先级阈值；在缓存中低优先级的消息，除非它的优先级被提升，否则不会被转发出去。因此，缓存大小对 PBQ 算法的整体传输量影响没有其它算法那么大。

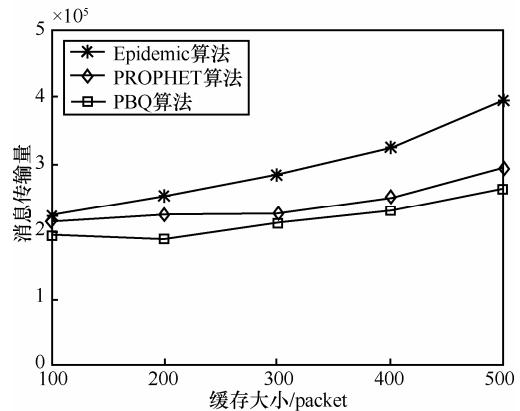


图 7 缓存大小对消息传输的影响

图 8 显示了缓存大小对各算法查询成功率的影响。可以看出，查询成功率随节点缓存的增大而提高。这是由于当缓存增大时，节点能同时存储较多的查询请求和结果消息，算法把这些查询消息发送到目标节点并成功返回的可能性也增大，因此提高了查询的成功率。同时，PBQ 算法在计算节点活跃系数时考虑了缓存剩余空间的因素，能避免活跃节点由于接收过多消息而导致缓存淘汰频率过高的问题。与其他 2 个算法相比，PBQ 算法的查询成功率提高了约 16%。通过设置较大的缓存空间，节点可以容纳更多的中转消息，算法减少了淘汰消息的数量，并提高了消息的递交率，进而提高查询的成功率。算法需要在冗余消息和查询成功率之间做一个平衡。

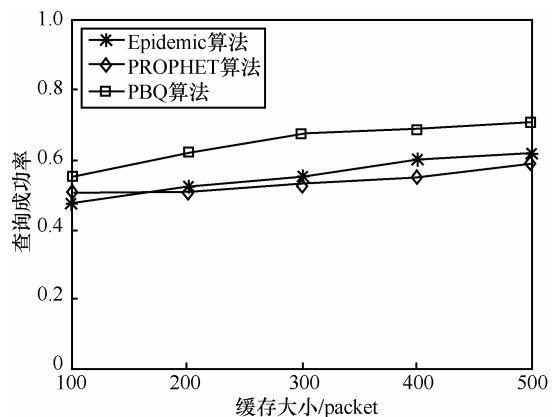


图 8 缓存大小对查询成功率的影响

最后, 定义查询请求百分比 RP (request percentage) 为查询请求的消息数与总消息数的比值, 进而分析算法中各阶段消息传输量的相对大小。|RE| 表示查询请求消息分组的数量, |RS| 表示查询结果消息分组的数量, 则 RP 定义为

$$RP = \frac{|RE|}{|RE| + |RS|} \quad (8)$$

如图 9 所示, 查询请求的百分比 RP 随着查询量的增大而增多。当查询量增大时, 有更多的查询请求消息往外扩散和传播, 但有些查询消息在中途会因某种原因被丢弃 (如因为缓存淘汰或未遇见可以回答查询的节点等), 导致查询结果返回阶段的消息传输量要少于查询请求阶段的消息传输量。然而, PBQ 算法的 RP 值比其他 2 个算法约高出 6%。这主要是因为 PBQ 在查询请求中包含了查询请求节点的相遇列表 CL。当查询结果返回时, 可根据该列表选择性地传输, 使消息传输时具有方向性, 从而使得网络中查询结果的消息量降低。根据式(8), 当查询结果的消息数|RS|降低时, 查询请求的消息数与总消息数的比值将增大。

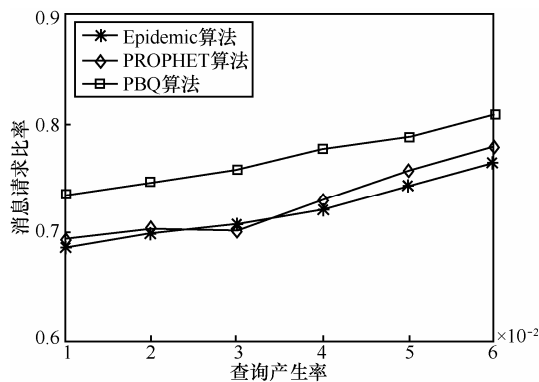


图 9 查询请求消息所占的百分比

6 结束语

随着机会网络和传感器网络技术的进一步融合, 机会无线传感器网络中的查询处理是一个关键性的问题。本文提出了机会无线传感器网络中一种基于优先级的查询分发和处理算法。算法分为初始运行、查询分发、查询执行与返回、查询清理等阶段。算法的基本思想是根据查询的优先级, 控制消息的分发和拷贝量; 并根据节点的活跃程度动态更新消息的优先级, 合理选取消息分发的中间节点; 在查询成功时清理网络中的查询残留消息, 从而优

化查询处理的时间, 提高查询处理的成功率和整体效用。实验结果表明, PBQ 算法能有效提高机会传感器网络中查询处理的成功率, 降低查询的开销和时间延迟。对于未来工作, 将根据不同类型查询的语义信息, 研究查询语义到消息优先级的映射方法; 同时, 利用节点的全局连接概率等元数据信息, 进一步研究查询优化的方法。

参考文献:

- [1] 熊永平, 孙利民, 牛建伟等. 机会网络[J]. 软件学报, 2009, 20(1): 124-137.
XIONG Y P, SUN L M, NIU J W, *et al.* Opportunistic networks[J]. Journal of Software, 2009, 20(1):124-137.
- [2] HULL B, BYCHKOVSKY V, ZHANG Y, *et al.* CarTel: a distributed mobile sensor computing system[A]. Proceedings of the 4th International Conference on Embedded Networked Sensor Systems[C]. ACM, Boulder, Colorado, USA. 2006. 125-138.
- [3] JUANG P, OKI H, WANG Y, *et al.* Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebraNet[J]. ACM SIGOPS Operating Systems Review, 2002, 36(5):96-107.
- [4] OCHIAI H, ISHIZUKA H, KAWAKAMI Y, *et al.* A dtn-based sensor data gathering for agricultural applications[J]. Sensors Journal, IEEE, 2011, (11):2861-2868
- [5] PENTLAND A, FLETCHER R, HASSON A. Daknet: rethinking connectivity in developing nations[J]. Computer, 2004, 37(1):78-83.
- [6] MADDEN S, FRANKLIN M J, HELLERSTEIN J M, *et al.* TAG: a tiny aggregation service for ad-hoc sensor networks[J]. ACM SIGOPS Operating Systems Review, 2002, 36(SI): 131-146.
- [7] LIU X, HUANG Q, ZHANG Y. Combs, needles, haystacks: balancing push and pull for discovery in large scale sensor networks[A]. Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems[C]. ACM, Baltimore, MD, USA, 2004. 122-133.
- [8] SCOTT K L, BURLEIGH S. Bundle protocol specification[EB/OL]. <http://tools.ietf.org/html/rfc5050>, 2007.
- [9] FALL K. A delay-tolerant network architecture for challenged internets[A]. Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications[C]. ACM, Karlsruhe, Germany, 2003. 27-34.
- [10] 张龙, 周贤伟, 王建萍等. 容迟与容断网络中的路由协议[J]. 软件学报, 2010, 21(10):2554-2572.
ZHANG L, ZHOU X W, WANG J P, *et al.* Routing protocols for delay and disruption tolerant networks[J]. Journal of Software, 2010, 21(10): 2554-2572.
- [11] VAHDAT A, BECKER D. Epidemic Routing for Partially Connected Ad Hoc Networks[R]. Duke University, 2000.
- [12] HARRAS A, ALMEROOTH K C, BELDING-ROYER E M. Delay tolerant mobile networks (dtmns): controlled flooding in sparse mobile

- networks[A]. Proceedings of the 4th TC6 International Conference on Networking Technologies, Services, and Protocols, Performance of Computer and Communication Networks, Mobile and Wireless Communication Systems[C]. Waterloo, Canada, 2005.1180-1192.
- [13] RAMANATHAN R, HANSEN R, BASU P, *et al.* Prioritized epidemic routing for opportunistic networks[A]. Proceedings of the 1st international MobiSys Workshop on Mobile Opportunistic Networking[C]. ACM, PR, USA, 2007. 62-66.
- [14] LINDGREN A, DORIA A, SCHELÉN O. Probabilistic routing in intermittently connected networks[J]. Service Assurance with Partial and Intermittent Resources, 2004.239-254.
- [15] 牛建伟, 周兴, 刘燕等. 一种基于社区机会网络的消息传输算法[J]. 计算机研究与发展, 2009,(12):2068-2075.
- NIU J W, ZHOU X, LIU Y, *et al.* A message transmission scheme for community based opportunistic network[J]. Journal of Computer Research and Development, 2009,(12):2068-2075.
- [16] BULUT E, SZYMANSKI B K. Exploiting friendship relations for efficient routing in mobile social networks[J]. IEEE Transactions on Parallel and Distributed Systems, 2012,23: 2254-2265.
- [17] CAO G G, PORTA T L, HAN J. On exploiting transient social contact patterns for data forwarding in delay tolerant networks[J]. IEEE Transactions on Mobile Computing, 2011,12(1):151-165.
- [18] HUI P, CROWCROFT J, YONEKI E. Bubble rap: social-based forwarding in delay tolerant networks[A]. Proceedings of the 9th ACM International Symposium on Mobile Ad hoc Networking and Computing[C]. ACM, 2008. 241-250.
- [19] ZHOU Y, LEVINE B N, CROFT W B. Distributed Information Retrieval for Disruption-tolerant Mobile Networks[R]. CIIR Technical Report IR-412, University of Massachusetts Amherst, 2005.
- [20] FAN J, CHEN J, DU Y P, *et al.* Delque: a socially-aware delegation query scheme in delay tolerant networks[J]. IEEE Transactions on Vehicular Technology, 2011, 60(5):2181-2193.
- [21] ZHAO Y, WU J. Socially-aware publish/subscribe system for human networks[A]. Proceedings of the Wireless Communications and Networking Conference[C]. Sydney, Australia, 2010.1-6.

作者简介:



赖永炫(1981-),男,福建龙岩人,厦门大学副教授、硕士生导师,主要研究方向为机会网络、传感器网络数据管理和数据库等。



林子雨(1978-),男,吉林柳河人,厦门大学讲师、硕士生导师,主要研究方向为数据库查询处理、实时主动数据仓库和数据挖掘等。