

文章编号:1001-9081(2008)01-0021-04

基于小世界模型的传感器网络节能查询策略

刘志强, 蒋泽军, 王丽芳, 王珺吉

(西北工业大学 计算机学院, 西安 710072)

(g. zqliu@gmail.com)

摘要: 查询处理是传感器网络中最重要的技术之一。以降低查询能耗为目的, 提出了一种基于小世界模型的查询策略(CardSN)。该策略以关联作为长程连接, 降低了网络平均路径距离; 将各节点的邻居相对定位, 使查询处理获得了方向性。仿真实验结果表明: CardSN 具有良好的可扩展性, 且能耗明显低于 ZRP 和 CAPTURE。

关键词: 查询; 小世界; 关联; 传感器网络

中图分类号: TP393.02 **文献标志码:** A

Small-world-based energy-efficient query strategy for sensor networks

LIU Zhi-qiang, JIANG Ze-jun, WANG Li-fang, WANG Jun-ji

(School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an Shaanxi 710072, China)

Abstract: Query processing is one of the most important technologies in sensor networks. To reduce energy consumption of query, a small world based query strategy named Contact-Assisted power-efficient Direction-sense-achieved query strategy for Sensor Networks (CardSN) was presented. In CardSN, Contacts act as shortcuts to bring down the average path length of the network; a distributed relative localization algorithm was introduced to achieve a sense of direction. Experimental results show that CardSN has good scalability and can save more energy than ZRP and CAPTURE. Therefore, CardSN is of high-performance and energy-efficiency.

Key words: query; small world; contact; sensor networks

0 引言

传感器网络以数据为中心的思想将整个网络视为分布式数据库^[1], 查询处理^[2]是在该数据库中获取感知数据的基本方法。因此, 查询策略算法的优劣是决定传感器网络系统性能的重要因素。同时, 传感器网络查询策略也是非常具有挑战性的课题, 这是因为: 1) 由于节点尺寸、成本等因素的限制, 使其存储、计算能力和通信带宽都有限, 因而无法实现集中的数据存储和查询; 2) 节点携带能量受限且无法补充, 因此要求查询机制高效节能; 3) 节点通常不安装 GPS 等定位装置, 因此节点不掌握地理位置信息和方向。

查询策略 MARQ^[3] 和 CAPTURE^[4] 依据小世界网络的研究成果, 令网络中的节点不仅掌握局部资源信息, 还分别与少数距离较远处的节点建立逻辑长程连接, 这些较远处的节点叫作关联节点(Contact)。网络利用节点掌握的局部资源和长程连接, 采用分布式查询算法, 实现全网的资源查询。上述策略具有高效节能、不存在瓶颈节点的特点, 因此应用前景广泛。但是, MARQ 和 CAPTURE 策略存在如下问题: 1) 选择关联节点时方向性较差, 造成源节点到关联节点的路径弯曲, 增大了查询能耗; 2) MARQ 策略受网络的运动模型约束较大, 不适用于节点移动性较弱的传感器网络资源查询; 3) CAPTURE 策略的关联节点选择是按需的, 存在大量重复关联选择能耗, 且查询时延较大。

本文针对上述策略的不足, 提出了一种查询策略——CardSN。CardSN 策略是辅助于关联节点、能量高效、能获得

方向性的传感器网络查询策略。

1 网络模型和问题描述

1.1 网络模型

本文论述内容限定在以下传感器网络系统模型范畴内, 多数传感器网络应用都符合下述特点:

- 1) 所有传感器节点均匀地部署在一个二维、边长为 Y 的监测区域 M 内, M 足够大。
- 2) 节点采用布尔感知模型, 即每个节点的感知范围是以节点为圆心、传感半径 R 为半径的圆, 在感知范围内的所有节点都可以接收到其发送的消息, 否则接收不到其发送的消息。
- 3) 节点的移动性较弱, 且是同构的。
- 4) 节点不配备 GPS 等任何能够确定节点地理位置、距离和方向的设备。

1.2 问题描述

传感器网络节点上感知或存储的数据(如温度测量值)称为资源。查询策略是某个节点获得其他节点上的资源的方法。CardSN 策略设计目的是在保证较高查询成功率的前提下, 实现对查询频率高的传感器网络的高效节能的资源查询。难点在于:

- 1) 源节点不知道目标资源的位置和方向;
- 2) 需要尽量降低通信能量消耗。

1.3 评价指标

综合文献[3, 4]的研究, 用以下标准评价策略性能:

- 1) 查询成功率(Query Success Rate)

收稿日期: 2007-07-23; 修回日期: 2007-10-16。 基金项目: 陕西省自然科学基金资助项目(2006F08)。

作者简介: 刘志强(1975-), 男(蒙古族), 内蒙古赤峰人, 博士研究生, CCF 会员, 主要研究方向: 传感器网络、复杂网络、网络存储; 蒋泽军(1964-), 男, 安徽合肥人, 教授, CCF 会员, 主要研究方向: 嵌入式系统、网络安全; 王丽芳(1964-), 女, 黑龙江齐齐哈尔人, 教授, CCF 会员, 主要研究方向: 嵌入式系统; 王珺吉(1976-), 男, 四川成都人, 博士研究生, 主要研究方向: 传感器网络。

查询成功率 = 能够查询到的节点数量 / 总节点数

2) 通信能耗 (Overhead)

通过网络中接收和发送控制消息的数量来评价, 消息数量决定网络负载和节点能耗。

2 CardSN 查询策略

2.1 理论依据

CardSN 策略以小世界模型^[5]作为理论依据。著名的 Stanley Milgram 实验发现, 通过平均 6 人次的熟人传递可以把社会中任意两个人联系起来, 这种现象称为小世界现象。小世界现象说明, 当网络呈现某种拓扑结构时, 仅利用局部信息就可实现高效的分布式信息搜索。WS 模型^[5]是典型的小世界模型: n 个节点分布在一个圆环上, 初始状态时, 每个节点有 k 个连接, 分别连向最近的 k 个节点。然后, 依次以概率 P 随机地改变各连接的终点。记 $D(i, j)$ 为节点 i 和 j 之间的最短距离, 平均路径距离 L 的计算公式如下:

$$L = \frac{2}{n(n-1)} \sum_{1 \leq i < j \leq n} D(i, j) \quad (1)$$

当 $P \approx 0$ 时, $L \sim n/2k$, 此时是规则网络拓扑。

当 $0.001 < P < 0.01$ 时, $L \sim \frac{\ln n}{\ln k}$, 此时节点不仅与相邻

节点存在连接, 还与远距离节点建立了少数的长程连接, 正是这些长程连接有效缩短了 L , 使整个网络呈现出小世界特征。网络的平均路径距离 L 越小, 资源查询效率会越高。

传感器网络节点只能与处在通信半径范围内的节点直接通信, 基本符合规则网络拓扑。如果要使传感器网络呈现小世界现象, 要在其中引入长程连接。长程连接有两种: 物理长程连接^[6]和逻辑长程连接^{[4]30}。CardSN 策略采用的是逻辑长程连接。

2.2 相关定义

定义 1 邻居。节点周围 H 跳以内的节点。对于任意节点 $q(x, y) \in M$, 其邻居定义为:

$$B(x, y) = \{ \forall (x', y') \in M \mid (x' - x)^2 + (y' - y)^2 \leq H^2 R^2 \}$$

其中: $H \geq 1$ 是正整数, 一般取值不能太大;

定义 2 边节点。恰好位于节点周围 H 跳上的邻居节点。定义为:

$$E(x, y) = \{ \forall (x', y') \in M \mid (H-1)^2 R^2 \leq (x' - x)^2 + (y' - y)^2 \leq H^2 R^2 \}$$

定义 3 近邻。能够与源节点直接通信的节点。

$$B'(x, y) = \{ \forall (x', y') \in M \mid (x' - x)^2 + (y' - y)^2 \leq R^2 \}$$

定义 4 关联节点最大距离 r 。关联节点距离源节点的最大限制跳数, 为了避免重叠, 令 $r \geq 2H$ 。

定义 5 最大关联节点数量 NoC 。每个源节点可以选择关联节点的最大数量。

定义 6 最大搜索深度 D 。源节点查询目标资源时, 搜索关联节点级数 (如关联的关联) 的最大值。

2.3 CardSN 查询策略原理

CardSN 策略的实现思想是: 通过在传感器网络中加入关联节点 (逻辑长程连接), 使其拓扑结构与小世界网络模型相符合。而后利用小世界现象实现高效节能的资源查询。基本实现步骤是:

1) 各节点选择并维护邻居信息和资源, 并周期性或触发性更新。

2) 各节点选择并维护关联节点 (不多于 NoC 个) 及源节

点到关联节点的路径。

3) 当某节点产生资源查询请求时, 执行资源搜索机制。该节点首先检索邻居资源是否存在目标资源 (以下简称 T)。如果存在, 查询成功; 否则, 则产生查询消息 (以下简称 QM), 将 QM 发送到所有关联节点。 QM 中包含: 源节点 ID、 T 和搜索深度 (以下简称 d_q), 初始时 $d_q = 0$ 。

4) 如果 QM 在传递过程中出现了断路, 则执行路径修复。如未修复成功则向源节点返回错误。

5) 当关联节点收到 QM , 首先进行查询环路和重复查询检查, 如果存在则抛弃 QM 。否则, 检索邻居资源中是否存在 T , 若是则返回查询结果; 否则, 令 QM 中的 $d_q = d_q + 1$, 而后转发至其关联节点。

6) 重复第 4 和 5 步, 若找到目标资源 T , 则查询成功; 若 d_q 达到 D 而未找到 T , 则查询失败。

图 1 描述了 CardSN 策略模型。CardSN 策略最重要的部分是使传感器网络的拓扑结构符合小世界网络模型拓扑。方法是通过长程连接改变网络的拓扑结构, 其中最重要的问题是如何选择并维护关联节点。

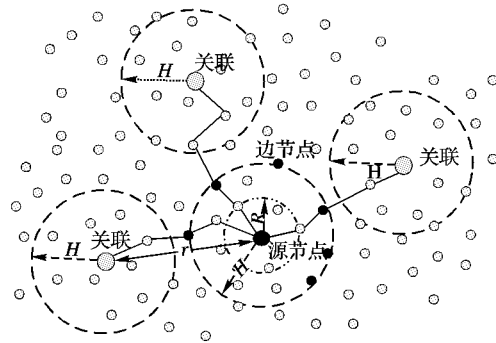


图 1 基于关联的查询策略概览

2.4 邻居信息选择与维护算法

在本策略中, 网络中的每个节点都保存邻居节点的资源, 并周期性和触发性地更新, 以保证资源的时效性。其中: 周期性更新的周期是预先设置的, 而触发性地更新发生在某个节点的资源发生变化时。例如, 通过传感器网络监测区域内某敌兵状态 (站立、匍匐或移动) 的应用中, 某节点感知到敌兵状态发生改变时, 或者由于节点失效、移动和新节点加入造成节点的近邻和边节点发生变化时, 将触发该节点的资源更新。需要更新的内容包括: 邻居节点感知的数据、邻居节点信息。具体算法如下:

1) 开始时, 网络中的每个节点首先与近邻组成一个无向图 $G = (V, E)$, 用邻接对称矩阵表示:

$$A[i, j] = \begin{cases} 1, & \text{当顶点 } V_i \text{ 和 } V_j \text{ 是近邻} \\ 0, & \text{当顶点 } V_i \text{ 和 } V_j \text{ 不是近邻} \end{cases} \quad (2)$$

2) 各节点将感知的资源数据、自身 ID 及近邻和边节点 ID 组成控制消息 (以下简称 CM), 设置 $ttl = H - 1$, 而后广播出去。

3) 节点收到 CM 后, 首先检查是否曾经接收过, 若是, 则抛弃掉 CM ; 否则, 检查是否 $ttl \geq 1$, 若是, 则令 $ttl = ttl - 1$, 而后将 CM 的副本广播出去, 并执行下一步; 否则, 直接执行下一步。

4) 节点对收到 CM 进一步处理: 将 CM 中的资源数据存储到资源列表; 同时, CM 源节点和近邻是该节点的邻居, 加入 $A[i, j]$, 并保存边节点。

5) 重复第 2 ~ 4 步, 直到 $A[i, j]$ 中的节点已收集完整。然后对 $A[i, j]$ 执行广度优先搜索算法生成邻居节点路径树, 并

确定边节点(距离根节点 H 跳的叶节点),保存边节点的边节点。邻居节点路径树记录了源节点到每个邻居的路径。

6) 确定边节点相对位置。

7) 节点周期性和触发性地执行第2~6步。

边节点的相对位置是指在边节点中任意选定一个节点作为“标志节点”,其他边节点通过边节点集合中的节点作为中继节点在一个“方向”上到达“标志节点”需要的跳数 t 。定位原理如图2(a)所示,节点2是节点1的近邻,节点3是节点2的近邻而不是点1的近邻,则节点3必定位于灰色区域内。依据该定位原理,定位算法如下:

1) 所有边节点组成集合 B ,从 B 中任选一节点作为“标志节点”,设其 $t = 0$ 。如图2(b)中节点1。

2) “标志节点”的所有近邻组成集合 C ,并设定变量 $C_i = 1$ 。令集合 $D_1 = B \cap C$ 中所有节点的 $t = C_i$,而后令 $C_i = C_i + 1$ 。如图2(b)中节点2和11的 $t = 1$ 。

3) 集合 D_1 中的每个节点的近邻分别组成集合 $W_i (i = 1, 2, \dots, x)$, x 是集合 D_1 中节点数目。令:

$$K = \bigcup_{i=1}^x W_i \cap B - D_1 \quad (3)$$

集合 K 中任取一个节点(例如:节点3)的近邻组成集合 F ,令:

$$D_2 = F \cap K \quad (4)$$

D_2 中的节点的 $t = C_i$,而后令 $C_i = C_i + 1$ 。这样就确定了相对位置的“方向”,接下来在这个方向上确定每个边节点与“标志节点”的相对位置。

4) 令集合 $D_{j-1} (j$ 为大于2的整数) 中各个节点的近邻分别组成集合 $W_i (i = 1, 2, \dots, x)$,令集合 D_j :

$$D_j = \bigcup_{i=1}^x W_i \cap B - D_{j-1} - D_{j-2} \quad (5)$$

D_j 中的节点的 $t = C_i$,而后令 $C_i = C_i + 1$ 。重复迭代执行本步骤,直到在 D_j 中出现“标志节点”时,算法结束。

最后结果如图2(b)所示的,所有边节点根据位置的不同,被划分为多个不同相对位置的集合。该过程中,由于空洞、网络边界等因素,使个别边节点不能够形成一个圆环,此时,要根据邻居节点路径树进行处理,限于篇幅,这里不作进一步论述。

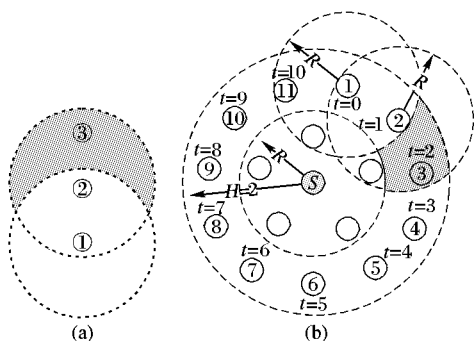


图2 CardSN 确定边节点相对位置的方法

确定边节点相对位置是为关联节点选择和资源搜索做准备。根据边节点的相对位置,可以做到:

1) 边节点扇区划分。是指根据边节点偏移量 t 的不同,将其划分为 $y \leq \max(t)$ 个基本相等的扇区,在第 i 个扇区节点偏移量 t_i :

$$\left\lfloor \frac{\max(t)}{y} (i-1) \right\rfloor \leq t_i < \left\lfloor \frac{\max(t)}{y} (i) \right\rfloor; i = 1, 2, \dots, y \quad (6)$$

其中: $\lfloor \cdot \rfloor$ 是指取整。

2) 选取中央节点。中央节点是指在一个连通边节点扇形区域内,基本处于扇形中间位置的边节点。例如,图2(b)中边节点11、1、2组成的边节点扇形区域中,节点1就是中央节点。

2.5 关联节点选择与维护算法

关联节点选择与维护是 CardSN 策略的核心,它的前提是节点掌握邻居信息。在选择关联节点时须尽量满足两个条件:1) 源节点邻居与它的任意关联节点的邻居之间不重叠;2) 相同源节点的不同关联节点的邻居之间不重叠。

节点的关联选择算法如下:

1) 节点将其边节点划分为 NoC 个扇区,若 $NoC > \max(t)$,则将其划分为 $\max(t)$ 个扇区。

2) 在任一扇区中任选一边节点 E 为目标节点,向其发送关联选择消息(以下简称 CS),CS 中包含下列信息:(1) 源节点的ID;(2) 源节点已选中的关联节点ID列表;(3) 节点 S 的方向边节点列表;(4) 距离源节点的跳数 d 。方向边节点列表包括:节点 E 的边节点集合与源节点的邻居集合的差集。

3) 当某节点收到 CS 时,将 d 加1。如果 d 值等于 r ,执行第5步。否则,将 CS 继续向下一跳传递,如果该节点已经是目标节点,则执行第4步。

4) 目标节点收到 CS,选取方向边节点列表所在扇形边节点区域内的中央节点作为下一个目标节点。然后将 CS 中方向边节点列表更新为新目标节点的边节点集合与当前节点的邻居集合的差集。最后将 CS 向下一个目标节点转发,执行第3步。

5) 检查该节点邻居中是否存在 CS 中保存的关联节点列表中的节点:如果不存在,该节点被选为源节点的关联,向源节点返回路径;如果存在,则向节点返回失败信息。

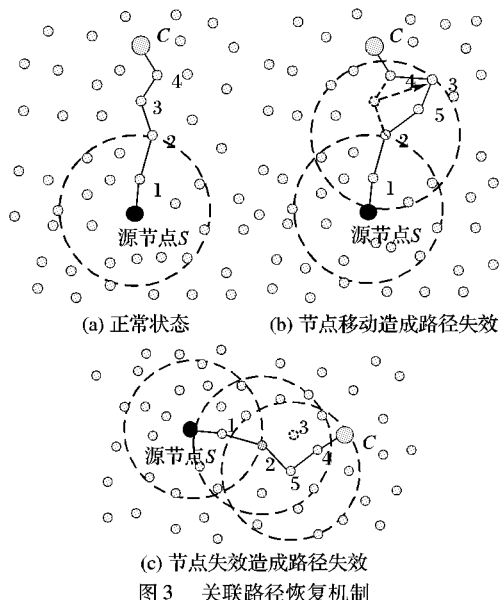


图3 关联路径恢复机制

6) 若源节点收到关联选择失败信息,则在当前扇区剩余未尝试节点中再任选一个,执行第2步,以此类推;若当前扇区所有节点都尝试过或源节点收到关联选择成功信息,则保存关联节点的路由,然后在剩余扇区中任选一个,并在该选中的扇区中任选一个边节点,执行第2步。依此循环,直到所有边节点都被尝试或者选中的关联节点数量达到 NoC 。

由于节点的失效、移动,可能会造成源节点到关联节点间的路径中断。采用路径恢复机制来恢复路径。路径恢复机制的具体方法如图3所示。图3(a)是正常的状态。图3(b)是

由于节点移动造成的路径中断:由于节点位置移动,节点 2 找不到其下一跳节点 3。节点 2 在其邻居中查找节点 3,找到后就可以获得到达节点 3 的路径,路径修复成功。图 3(c)是由于节点失效造成的路径中断:节点 2 在其邻居中找不到失效的节点 3,则节点 2 在其邻居中查找能直接到达节点 C 的邻居内节点,因而发现节点 5,路径修复成功。

2.6 资源搜索机制及环路和重复查询的避免

CardSN 的搜索策略包括两个过程:路径查询阶段和路径回复阶段。路径查询阶段采用迭代加深方法 (Iterative Deepening), 基本思想是给定最大搜索深度 D , 源节点在其邻居中不存在目标资源的情况下,以关联节点的级数递增的次序,依次查询关联节点的邻居中是否有目标资源。即依次查询第 1 级关联节点,第 2 级关联节点(关联节点的关联节点),……,第 i 级关联节点,其中 $1 \leq i \leq D$,直到查找到目标资源或者达到最大搜索深度 D 。这种机制叫作逐级搜索 Ibl 机制。如果每次查询都执行深度为 D 的搜索,则叫作单次搜索 single-shot 机制。如果搜索深度每次以倍数增长($d_q = Cd_q$, $C \in \mathbf{N}^+$),称为分步搜索 step 机制。

路径回复阶段指采用上述机制找到目标资源后,将其返回到源节点。

查询消息 QM 在关联节点中多级转发过程中,可能会出现查询环路或相同的区域被重复查询的情况。重复查询和查询环路会造成大量无意义的通信能耗。采用如下方法减少这种现象:

- 1) 当 QM 的下一跳节点与前一跳节点 P 相同时,说明 QM 开始沿原路回传,阻断 QM 的传递;
- 2) 在 QM 中加入序列号 SN,它能够唯一的标识 QM;QM 源节点 Q ,是指发出该 QM 的节点 ID。如果某个关联节点两次或两次以上收到相同 SN 和 Q 的 QM 时,说明出现了查询环路,则中断该分支路径的查询。

通过上述手段将有效减少重复查询和查询环路的出现。当然,并不能杜绝重复查询和查询环路。

3 CardSN 策略仿真分析评估与比较

与其他相关策略相比,CardSN 策略的优势在于:1) 关联节点的引入使网络的拓扑基本符合小世界网络模型的拓扑,实现了高效的分布式信息搜索;2) 在不需要节点地理位置信息的前提下,通过定位算法,获得边节点的相对位置,提高了选择关联节点的质量和搜索资源的效率。

策略的通信能耗主要包括三部分:邻居选择与维护能耗 E_{vicinage} 、关联节点选择与维护能耗 E_{Contact} 和单次查询能耗 E_{query} 。用 n 表示一个邻居资源更新周期内的查询次数,则每次查询平均查询能耗:

$$E_{\text{avg}} = \frac{E_{\text{vicinage}} + E_{\text{Contact}} + nE_{\text{query}}}{n} \quad (7)$$

由式(7)可以看出, n 值越大, E_{avg} 越小,此时 E_{vicinage} 和 E_{Contact} 在 E_{avg} 中占的比例很低。也就是说,本策略适用于查询频率较高的网络。下面根据 1.3 节的指标,通过仿真来分析它的性能,并与相关策略进行比较。

3.1 CardSN 策略参数值对查询成功率的影响

采用 NS-2 及其扩展程序作为仿真实验平台,场景 500 节点,均匀分布在边长为 1400 m 的方形面积上,节点的通信半径是 110 m。图 4 是实验结果,反映了参数选择对查询成功率的影响。生成图 4 各条曲线的网络参数选择如下:1) R 对查

询成功率的影响曲线: $r = 2R + 1, D = 1, NoC = 4$;2) r 对查询成功率的影响曲线: $R = 3, D = 1, NoC = 4$;3) NoC 对查询成功率的影响曲线: $r = 2R + 1 = 7, D = 1$;4) D 对查询成功率的影响曲线: $r = 2R + 1 = 7, NoC = 4$ 。参数值的选取是根据文献[4]的研究经验选取的。

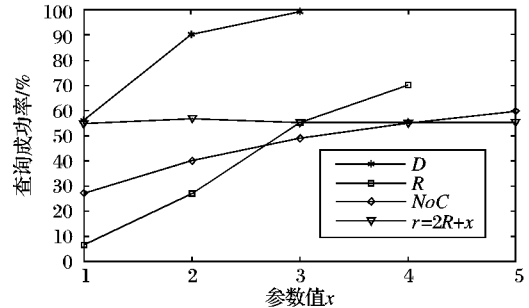


图4 参数对查询成功率的影响

由图4得出如下结论:随着 R 、 NoC 和 D 值的增大,查询成功率迅速提高,其中, D 值对查询成功率的影响最大。参数 r 对查询成功率的影响较小。

3.2 CardSN 策略参数值对通信能耗的影响

采用与 3.1 节相同的仿真场景,参数选择如下:1) R 对能耗的影响曲线: $r = 7, D = 2, NoC = 6$;2) r 对能耗的影响曲线: $R = 3, D = 2, NoC = 6$;3) NoC 对能耗的影响曲线: $r = 2R + 1 = 7, D = 2$;4) D 对能耗的影响曲线: $r = 2R + 1 = 7, NoC = 4$ 。

由图5可以得出:随着 R 、 NoC 和 r 值的增大,能耗上升。由图6可以得出:随着 D 值的增大,能耗迅速上升,上升幅度远大于其他三个参数。

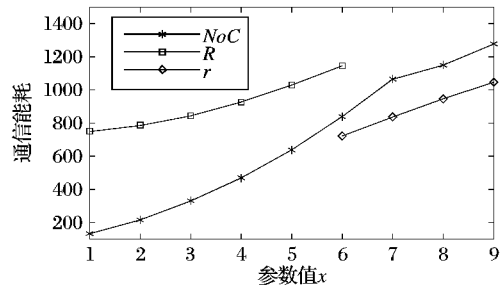


图5 参数对能耗的影响

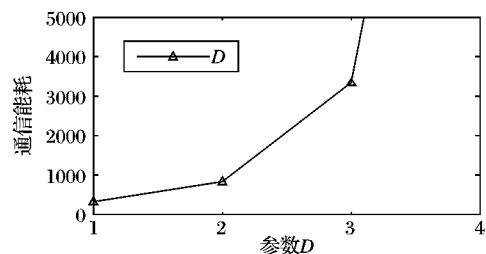


图6 参数D对能耗的影响

上述试验是参数取值对查询成功率和能耗的影响趋势的分析。从仿真试验可以得出,随着参数 R 、 r 、 NoC 和 D 取值的增大,查询成功率上升,但同时通信能耗也上升。因此,选择参数值时要兼顾查询成功率和通信能耗。

3.3 CardSN 能耗仿真分析与比较

本节将 CardSN 策略能耗与 CAPTURE、ZRP^[7] 和 Flooding 协议进行比较分析。表1是仿真场景和 CardSN 策略参数选择。参数选择原则是保证在不同场景中的查询成功率高于 90%。每个节点随机选择5个目标节点,执行查询操作,将总

(下转第51页)

越小,而域内切换的概率越大,平均切换时间越小,这正是改进切换机制支持节点微移动的优势所在。当 $p_{\text{inter}} = 0.84$ 时,平均切换时间可以下降接近 10%,而 $p_{\text{inter}} = 0.5$ 时,平均切换时间可以下降 20% ~ 30%; $p_{\text{inter}} = 0.28$ 时,可以下降 30% ~ 50%。 ε 越大,表明外地代理与通信对端相距越远(对应移动节点与通信对端相距越远),平均切换时间越小,因为扩展域后,域内移动仅需要进行域内切换,而域内切换时间变小,因此平均切换时间就减小,由图 4 可以看出这种趋势,当 $p_{\text{inter}} = 0.28$, $\varepsilon = 0.6$ 时平均切换时延仅为先验式的 73%,而 $\varepsilon = 2.1$ 时,平均切换时延仅为先验式切换时间的 51.23%。切换的其他性能参数诸如切换分组丢失和切换时应用吞吐量都与切换时间相关,因此随着切换时间的减小,分组丢失数量将得到有效的减小,吞吐量的下降将得到缓解。

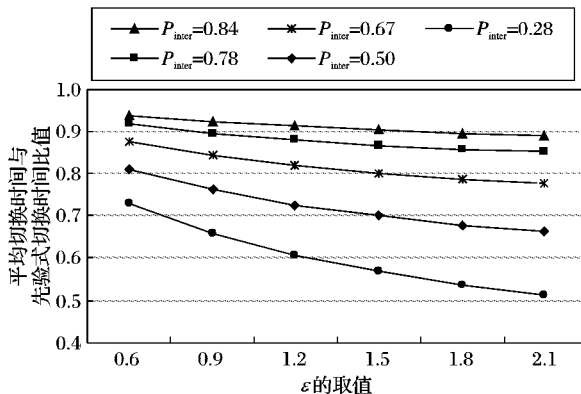


图 4 平均切换时间与先验式切换时间比值随 ε 的变化

(上接第 24 页)

的通信量对总查询次数取平均值作为实验结果。

表 1 仿真场景(节点通信半径 110 m)

节点数	面积/m ²	R	r	NoC	D
200	1000 × 1000	3	7	4	2
500	1400 × 1400	3	7	4	3
1000	2000 × 2000	3	7	6	3
2000	2800 × 2800	3	7	6	4
4000	3700 × 3700	3	7	6	5

图 7 是仿真结果,其中纵轴是 CardSN、CAPTURE、ZRP 协议能耗与 Flooding 协议能耗的比值。CAPTURE、ZRP 协议的实验数据取自文献[4]。由图 7 得出: CardSN 的能耗比 CAPTURE 低 15% 左右,比 ZRP 协议低 18% 以上。虽然不同的场景、节点分布及参数选择将带来不同的能耗,但是,该实验充分证明了 CardSN 在保证查询成功率的前提下,在能耗上优于 CAPTURE 和 ZRP 协议。

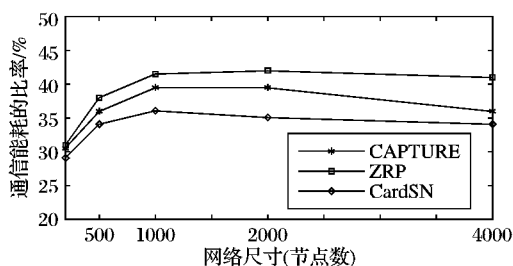


图 7 CardSN、CAPTURE 和 ZRP 能耗比较

4 结语

CardSN 策略以关联节点作为长程连接,并在不需要节点

4 结语

Ad Hoc 网络因其不需要固定基础设施、组网方便等特性,得到了越来越多的应用,但是单独的 Ad Hoc 网络应用比较特殊,它作为末端网络接入 Internet 是其发展的趋势,节点的移动性必然带来切换管理,但是采用标准移动 IPv6 切换机制存在许多问题。本文通过在边界路由器和子网网关中新增外地代理和本地代理,将边界路由器内的多个子网扩展为一个域,将域内的移动对网络和通信对端屏蔽,从而达到支持微移动性的目的。分析表明节点在域间移动概率较小、移动节点远离通信对端的情况下改进的切换机制可以大大减小平均切换时间。

参考文献:

- [1] JOHNSON D, PERKINS C, ARKKO J. Mobility support in IPv6, Draft-ietf-mobileip-ipv6-17. txt[R]. IETF, 2002.
- [2] LEI H, PERKINS C. Ad Hoc networking with mobile IP[C]// Proceedings of the Second European Personal Mobile Communications Conference(EPMCC 97). Bonn: [s. n.], 1997: 197-202.
- [3] 聂晶. 下一代网络中的移动性管理[D]. 北京: 北京邮电大学, 2005.
- [4] JONSSON U, ALRIKSSON F, LARSSON T, et al. MIPMANET - mobile IP for mobile Ad Hoc networks[C]// The First IEEE/ACM Annual Workshop on Mobile Ad Hoc Networking and Computing (Mobi-HOC 2000). Boston: IEEE Press, 2000: 75-85.
- [5] 薛蓉蓉. 移动 IP 中位置管理机制的分析及优化研究[D]. 长沙: 湖南大学, 2004.

位置信息的前提下,使查询具备方向性,提高了搜索效率。仿真试验结论表明,CardSN 策略是高效节能的查询策略。如何将 CardSN 策略应用于传感器网络数据库查询解析操作中或者路由协议中是下一步的研究方向。

参考文献:

- [1] YAO Y, GEHRKE J. The cougar approach to in-network query processing in sensor networks [J]. SIGMOD Record, 2002, 31(3): 9-18.
- [2] GEHRKE J, MADDEN S. Query processing in sensor networks [J]. IEEE Pervasive Computing, 2004, 3(1): 46-55.
- [3] HELMY A. Mobility-assisted resolution of queries in large-scale Mobile Sensor Networks (MARQ) [J]. Computer Networks Journal, 2003, 43(4): 437-458.
- [4] HELMY A. CAPTURE: Location-free contact-assisted power-efficient query resolution for sensor networks [J]. ACM SIGMOBILE Mobile Computing and Communications Review, 2004, 8(1): 27-47.
- [5] WATTS D J, STROGATZ S H. Collective dynamics of 'small-world' networks [J]. Nature, 1998, 393: 440-442.
- [6] SHARMA G, MAZUMDAR R. Hybrid sensor networks - A small world [C]// Proceedings of the 6th ACM international symposium on Mobile Ad Hoc networking and computing. Urbana-Champaign: ACM Press, 2005: 366-377.
- [7] HAAS Z J, PEARLMAN M R. The performance of query control schemes for the zone routing protocol [J]. IEEE Transactions on Networking, 2001, 9(4): 427-438.
- [8] 郭龙江, 李建中, 李贵林. 无线传感器网络环境下时一空查询处理方法[J]. 软件学报, 2006, 17(4): 794-805.
- [9] 张细政, 胡忠望, 肖建华, 等. 无线传感器网络中基于消息预取的移动查询技术[J]. 计算机应用, 2006, 26(11): 2531-2535.