

WSN 中基于树型标号系统的分布式路由算法

侯贵升, 吴晓蓓

(南京理工大学自动化学院, 南京 210094)

摘 要: 针对无线传感器网络中数据中心存储的路由问题, 提出一种基于树型标号系统的分布式路由算法。将网络中的节点组织成以参考节点为根的树型结构, 通过比较目的节点标号与邻居节点标号, 选择转发节点, 实现数据路由。分析与仿真结果表明, 该路由算法的空间开销较低、路由效率较高, 并且生成的路径接近最短路径。

关键词: 无线传感器网络; 数据中心存储; 树型标号系统; 路由; 空间开销

Distributed Routing Algorithm Based on Tree Label System in WSN

HOU Gui-sheng, WU Xiao-bei

(School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China)

【Abstract】 Aiming at routing problems of Data-centric Storage(DCS) in Wireless Sensor Network(WSN), this paper proposes a Tree Label System(TLS) based distributed routing algorithm. The algorithm organizes network nodes into a spanning tree whose root is a certain reference node in the network. For data routing, it only needs to compare the destination node's label with neighbor nodes' labels and select appropriate forwarding nodes based on the results. Analysis and simulation results show that the algorithm routing space cost is low, and its routing efficiency is high, and its generated path is close to the shortest path.

【Key words】 Wireless Sensor Network(WSN); Data-centric Storage(DCS); Tree Label System(TLS); routing; space cost

DOI: 10.3969/j.issn.1000-3428.2012.21.020

1 概述

无线传感器网络(Wireless Sensor Network, WSN)集感知、计算、通信为一体^[1], 它无需固定基础设施支持, 具有低成本、易部署、容错性高、抗毁性强等特点, 应用前景十分广泛。WSN 以数据为中心, 感知数据的存储方式是 WSN 研究中的一个重要问题^[2]。已有研究表明: 数据中心存储(Data-centric Storage, DCS)是 WSN 尤其是大规模 WSN 中数据存储的首选方式^[3-5], 它具有能耗低、通信量小以及分布式的特点。但是, DCS 的路由问题却是一个难点^[6]。文献[3-5]分别使用节点的物理坐标和虚拟极坐标实现数据路由, 前者要求每个节点配备一个定位装置, 因而网络成本较高; 后者的虚拟极坐标表示较复杂, 导致节点空间开销偏高。两者皆不适用于较大规模 WSN。

为此, 本文在文献[5]的基础上设计了一种基于树型标号系统(Tree Label System, TLS)的分布式路由算法 LBR。LBR 的主要思想是将网络中的节点组织成一个以网络中心某参考节点为根的生成树并为树中每个节点分配一个隐含其相对位置信息的标号, 从而建立一个以参考节点标号为根的树型标号系统, 实现网络拓扑到标号空间的映

射, 从而使节点间的数据路由通过对其标号的运算实现。

2 TLS 标号系统

网络拓扑需要映射到标号空间以支持 LBR 路由, 感知数据名的集合也需映射到标号空间以实现数据中心存储, 因此, 标号系统是基础, 也是关键。标号系统基于以下假设:

- (1) 每个节点有唯一标识或编号, 如节点 ID。
- (2) 节点间的通信链路为双向。
- (3) 指定参考节点, 并尽量部署在网络中心区域。

2.1 标号系统的建立

标号系统的建立过程由节点状态控制, 节点状态包括游离(标号为空且无父节点)、等待(标号为空但有父节点)和正常(标号不为空)3 种, 初始时节点均处于游离状态, 节点的状态转换由消息驱动。整个建立过程如下:

(1) 参考节点将自身标号置为 0, 进入正常状态, 并广播 iMsg 从而启动建立过程。

(2) 游离节点(处于游离状态的节点)收到 iMsg 后, 以其源节点为父节点, 进入等待状态, 并向其发送 askMsg。

(3) 正常节点收到 askMsg 后, 将其源节点加入子节点

基金项目: 教育部博士点基金资助项目(20113219110028)

作者简介: 侯贵升(1985—), 男, 博士研究生, 主研方向: 无线传感器网络, 路由协议; 吴晓蓓, 教授、博士生导师

收稿日期: 2011-12-01 **修回日期:** 2012-03-09 **E-mail:** guisheng_hou@163.com

集, 并按照加入的先后次序从 0 开始依次为其分配序号; 由于子节点数不得超过其上限值, 因此后加入的节点将可能被拒绝, 分配结束后, 结果通过 allocMsg 广播出去。

(4) 等待节点收到 allocMsg 后, 先确认是否被拒绝, 若被拒绝, 则重新回到游离状态, 否则从中取出自己的序号(特征序号)并将其接在父节点标号后作为自身标号, 同时将自身状态设为正常, 然后广播 iMsg 以推进建立过程。

当网络中节点的状态都转为正常后, 标号系统建立成功, 如图 1 所示。在正常情况下, 绝大多数节点只需发送 3 个消息: askMsg, allocMsg 以及 iMsg, 由此引入的额外能量开销和通信开销均不高。若节点总数为 n , 则对于标号系统的建立, 其算法复杂度为 $O(n)$; 网络边缘节点标号的长度随生成树的深度增加而增加, 其空间复杂度为 $O(\log n)$ 。

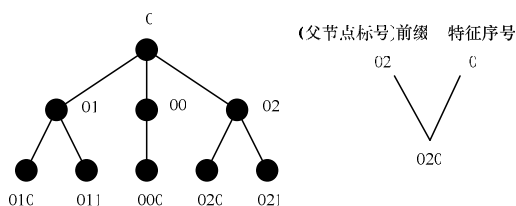


图1 TLS标号系统结构

2.2 一致性条件约束

为使 DCS 和 LBR 能正确、高效地运行, 标号系统及其反映的网络结构须满足以下条件约束:

- (1) 每个节点须有一个父节点(根节点除外)。
- (2) 节点标号由其特征序号接在父节点标号后生成。
- (3) 同父节点特征序号不同。

显然, 采用 2.1 节描述的算法所构建的树型网络结构以及相应的标号系统(TLS)满足以上条件约束。因此, TLS 具有支持数据中心存储和点到点路由的特点:

(1) 支持数据中心存储: 设某一感知数据名为 k , 通过映射函数 f , 该数据将被存储到标号为 $f(k)$ 的存储节点上; 相应地, 一个对该项数据的请求按照同样的过程会被定位到该节点。

(2) 支持点到点路由: 在 LBR 中, 节点欲与非邻居节点通信必须知道对方的标号, 若对方是某项数据的存储节点, 通过映射函数或哈希表, 节点能轻易获得; 若不是, 则需借助 DCS 向附近的存储节点查询, 前提是存储节点有该节点的通信记录, 否则, 只能采用洪泛查询方式获得。

2.3 标号系统的维护

随着网络的运行, 网络中的某些节点可能因能量耗尽或其他外部因素而失效, 这将导致网络出现“监测盲区”。为此, 需及时向网络中加入新节点以接替失效节点继续工作。无论是失效节点的退出还是新节点的加入都可能会破坏既有标号系统和一致性条件约束, 从而影响网络的正常运行, 因此需要相应的维护措施, 如图 2 所示, 其中, P 子树是指以 P 为根的子树; askMsg 和 ansMsg 的格式如表 1 所示。

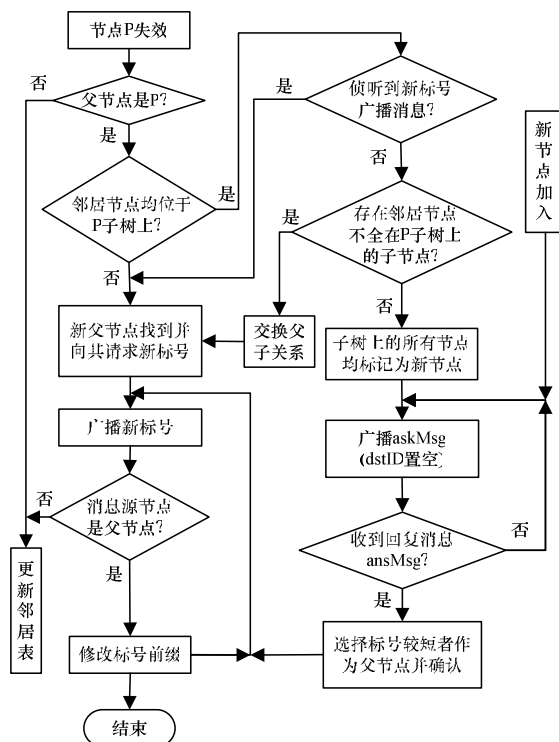


图2 TLS标号系统维护流程

表1 主要消息列表

消息名	格式
iMsg	srcID label
askMsg	srcID dstID
allocMsg	srcID label ID[n]
ansMsg	srcID dstID label

在表 1 中, srcID、dstID 分别表示源节点和目的节点 ID; ID[n] 表示子节点 ID 数组, n 既表示数组索引也表示相应子节点的特征序号。

在标号系统的维护过程中, 应该尽量减少需调整标号的节点以维持系统的稳定性。新节点加入和叶节点失效均不会影响其他节点, 但非叶节点失效, 则会导致其子孙节点调整标号。一般而言, 对于均匀、密集部署的网络, 网络中的节点可近似看作围绕网络中心按距离其跳数呈同心圆环分布。假设网络半径为 10 跳, 1 个节点失效将导致平均约 6 个节点调整标号, 但其中仅 1 个需完全调整其标号, 而其余 5 个只需修改其标号前缀。小范围内调整节点标号对 LBR 的影响可忽略, 但若涉及范围较大时, 则应进行局部重建。

3 基于树型标号系统的路由算法

利用邻居表和目的节点标号, LBR 能轻易实现点到点路由。根据邻居表的组织情况, LBR 可细分为 LBR-UD (LBR for Up and Down Forwarding) 和 LBR-EX (LBR for Extra links Forwarding)。前者仅使用节点间的父子关系, 只能向上或向下转发数据, 而后者还使用其他关系比如兄弟、叔侄等关系, 能将数据从任何一个有链路连接的方向转发出去。显然, LBR-EX 相对于 LBR-UD 路由效率更高, 所选路径更接近最短路径, 但操作较后者复杂且空间开销

比后者大很多。LBR 是通过标号间的关系路由的,下面是相关的一些定义:

定义 1(相同) 长度相同且相应位均相等的两标号相同或相等,否则不同。

定义 2(包含) 两不同标号 a 与 b , a 的长度小于 b 且 a 的相应位与 b 的对应位均相等,则称 a 包含 b 。

在 2.1 节所述的 TLS 中,标号相同的两节点必为同一节点,否则为不同节点;若两节点标号为包含关系,那么被包含者必位于以包含者为根的子树上。

在已有的一些研究中^[7],生成树被广泛用于从网络中的普通节点到基站的路由,生成树均以基站为树根,而且普通节点一般以最小跳数与基站通信。利用 TLS,该类算法很容易扩展为支持点到点的路由算法:源节点产生的消息先沿着树向上,到达目的节点的某一祖先节点后,再顺着树向下直抵目的节点。途经每一节点时,当前节点只需比较自身标号(SL)与目的节点标号(DL)并根据其结果转发消息,若两者相同,则消息到达目的节点;若 SL 包含 DL,则将消息转发给标号包含或等于 DL 的子节点;否则,交给父节点,如图 3(a)所示。

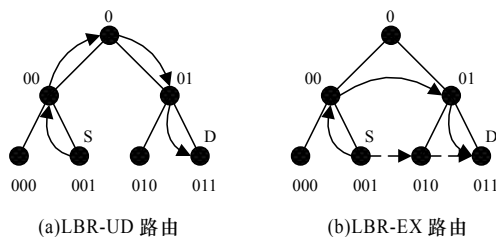


图 3 LBR 路由示意图

LBR-EX 与 LBR-UD 路由的基本过程类似,不同点在于:当前节点在将消息向上转发给父节点之前需检查除父子节点外的躯体邻居节点,看是否存在包含目的节点标号或与之相同的标号,若存在,则将消息转发给对应的节点,否则,仍交给其父节点。如图 3(b)中实线路径所示, S 到 D 的跳数相对于使用 LBR-UD 路由节省了 1 跳。

LBR-EX 可进一步加以改进,只需让每个节点维护一个 2 跳邻居表。此时,节点拥有更开阔的视野,因而数据路由时当前节点更可能在其邻居中找到合适的下一跳节点而无需将消息向上交给其父节点,从而缩短到达目的节点的路径并显著提高路由效率,如图 3(b)中虚线路径所示, S 到 D 仅需 2 跳即到达,比使用 1 跳邻居表节省 1 跳,并且 S 到 D 的路径为最短路径。由此可知,若每个节点都维护一个 n 跳邻居表,路由效率将随着 n 的增大而提高,但同时邻居表的大小将呈指数膨胀(如果不采用有效压缩策略)。

4 仿真分析

使用基于 OMNET++^[8]的 WSN 仿真架构进行仿真实验:网络拓扑结构为树型,汇聚节点作为根节点部署在正方形网络区域的中心,其余若干传感器节点随机均匀分布在区域内各处,节点的平均邻居节点数为 15;物理层和

MAC 层分别采用理想无线收发器和理想 MAC 协议,即如果两节点间的距离小于通信距离,那么其间通信的丢包率为 0;设定节点间的通信距离为 30 m;设置 20 种感知数据类型,分别对应 20 个标号,感知数据类型由源节点随机决定,源节点每秒发送 50 个数据包,仿真时间为 50 s;LBR-EX 使用 2 跳邻居表;每种仿真设置分别在 5 个随机产生的网络拓扑上各运行 1 次,然后取其平均值作为该设置下的仿真结果。下面主要从路由效率和空间开销 2 个方面对 LBR 进行评估,并与文献[3]中提出的 GEM 算法作比较。

4.1 不同网络规模下的平均路径长度比

平均路径长度比即实际路径与最短路径长度比的平均值,其值越小,路由效率越高。图 4 显示了平均邻居节点数为 15 时不同网络规模下 LBR-UD、LBR-EX 以及 GEM 的平均路径长度比。由于三者的路由选择皆基于局部拓扑信息,因此随着节点数的增加,网络区域的增大,实际路径偏离最短路径的可能性增大,其平均路径长度比相应增加;而 GEM 将网络中的节点组织为带环的树型结构,同级节点(距离根节点跳数相同的节点)的角度范围依据其地理位置顺序递增或递减,由于其路由效率对于网络区域的增大并不敏感,因此其平均路径长度比增加相对较平缓。LBR-UD 只向上或向下路由数据包,由此,平均路径长度比较大,而 LBR-EX 在所有邻居中选择最合适的节点作为下一跳节点,生成的路径在网络区域不是很大时接近最短路径。GEM 虽为贪婪算法,但因其倾向于选择同辈节点作为转发节点,以充分利用环状树的环带链路,当目的节点与源节点角度范围跨度较大时,生成路径的弧线特征明显,长度随之增加。因此,相对于使用 2 跳邻居表的 LBR-EX,其平均路径长度比偏高。

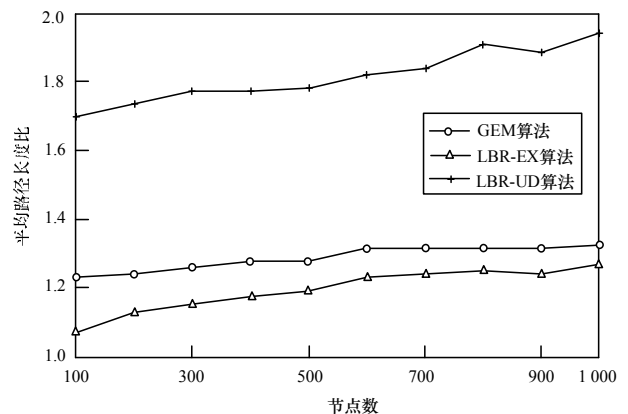


图 4 不同网络规模下的平均路径长度比 1

4.2 不同节点密度下的空间开销

图 5 显示了网络区域为 $300\text{ m} \times 300\text{ m}$ 时不同节点密度下 GEM 的节点最大空间开销(GEM max)、平均空间开销(GEM avg), LBR-UD 的节点最大空间开销(LBR-UD max)、平均空间开销(LBR-UD avg)和 LBR-EX 的节点最大空间开销(LBR-EX max)、平均空间开销(LBR-EX avg)。节点密度增大,即平均邻居节点数增大,邻居表相应扩展,节点

空间开销随之增加。GEM 用<跳数,角度范围>表示网络中的节点, 不仅操作复杂, 而且邻居表占用空间较大。在 LBR 中, 节点标号由一系列特征序号串接而成(见 2.1 节), 而特征序号都很小, 小于节点子节点数上限值, 该上限值随距离根节点的跳数增加而递减, 同时, 邻居表依据树型网络结构的特点进行压缩存储: 按邻居节点与自身的关系组织邻居表, 这些关系包括父子、兄弟、叔侄和其他共 6 种关系, 除其他外其余 5 种关系中的节点都只需存储其特征序号即可, 因此, 空间开销较低。LBR-UD 只使用节点间的父子关系, 其邻居表只需存储父节点和子节点的特征序号, 因此, 空间开销低于 LBR-EX。

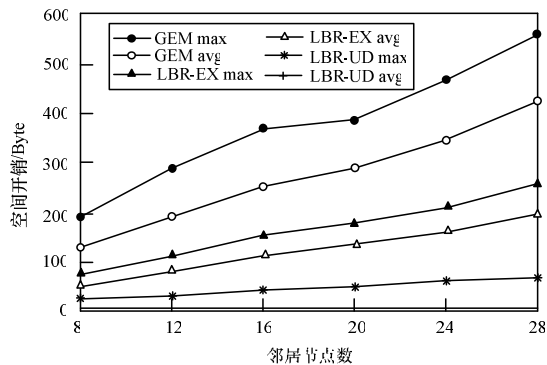


图5 不同节点密度下的空间开销

4.3 LBR-EX 的最佳邻居表跳数

LBR-EX 的路由效率与使用的邻居表的跳数有关: 跳数越大, 节点掌握的网络拓扑信息越全面, 生成的路径越接近最短路径, 因此, 路由效率越高, 但空间开销相应地也越大。图 6 和图 7 分别显示了平均邻居节点数为 15 时不同网络规模下采用不同跳数邻居表的 LBR-EX 的平均路径长度比和相应的节点平均空间开销。可以看出, 采用 2 跳邻居表(LBR-EX2)相对于采用 1 跳邻居表(LBR-EX1), 其平均路径长度比大大降低, 路由效率大幅提升; 而 3 跳邻居表(LBR-EX3)、4 跳邻居表(LBR-EX4)相对于 2 跳而言, 路由效率虽有不小提升, 但节点平均空间开销的增长却十分迅速。因此, 对于 LBR-EX, 在一般情况下, 选择 2 跳邻居表能达到路由效率与空间开销间的最佳平衡。

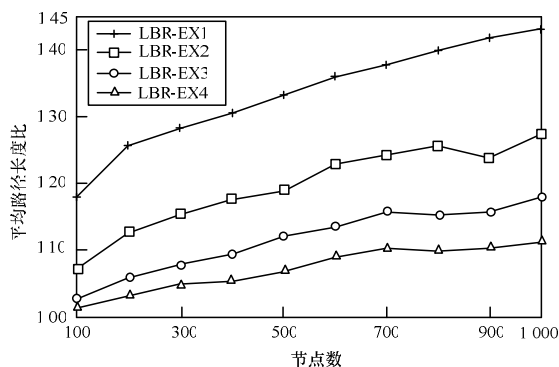


图6 不同网络规模下的平均路径长度比 2

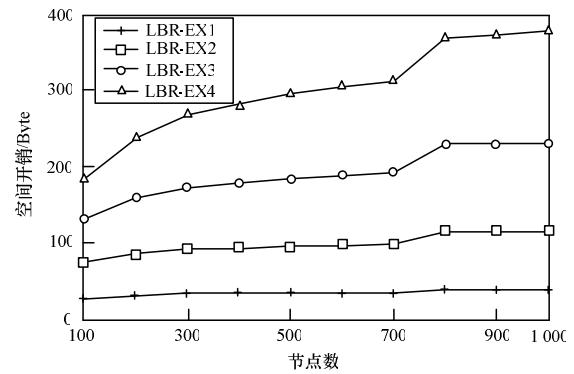


图7 不同网络规模下的空间开销

5 结束语

为解决 WSN 中数据中心存储的路由问题, 本文提出一种基于树型标号系统 TLS 的分布式路由算法。该算法将网络中的节点组织为一个以参考节点为根的生成树, 并借助图标号技术为每个节点分配一个唯一且包含其相对位置信息的标号, 使节点间的数据路由通过节点标号间的运算关系实现。算法简便可行、路由效率高、空间开销低, 但只适于节点静止的网络, 因此, 使本文算法支持移动性网络将是下一步研究方向。

参考文献

- [1] 孙利民, 李建中, 陈渝, 等. 无线传感器网络[M]. 北京: 清华大学出版社, 2005.
- [2] 蔚赵春, 周水庚, 关伟红. 无线传感器网络中数据存储与访问研究进展[J]. 电子学报, 2008, 36(10): 2001-2010.
- [3] Ratnasamy S, Karp B, Li Yin, et al. GHT: A Geographic Hash Table for Data-centric Storage[C]//Proc. of WSNA'02. Atlanta, USA: ACM Press, 2002: 78-87.
- [4] Shenker S, Ratnasamy S, Karp B, et al. Data-centric Storage in Sensornets[J]. ACM SIGCOMM Computer Communications Review, 2003, 33(1): 137-142.
- [5] Newsome J, Song D. GEM: Graph Embedding for Routing and Data-centric Storage in Sensor Networks Without Geographic Information[C]//Proc. of the 1st ACM Conference on Embedded Networked Sensor Systems. Redwood, USA: ACM Press, 2003: 76-88.
- [6] Fang Qing, Gao Jie. Landmark-based Information Storage and Retrieval in Sensor Networks[C]//Proc. of INFOCOM'06. Barcelona, Spain: IEEE Press, 2006: 1-12.
- [7] Sam M, Franklin M J, Joseph M. TAG: A Tiny Aggregation Service for Ad Hoc Sensor Networks[C]//Proc. of the 5th Symposium on Operating Systems Design and Implementation. New York, USA: [s. n.], 2002: 131-146.
- [8] OMNET++ Community[EB/OL]. (2010-08-17). <http://www.omnetpp.org>.

编辑 陆燕菲