

文章编号:1671-6833(2011)03-0111-04

## 一种无线传感器网络分层拓扑推断算法

申金媛, 赵旭东, 刘润杰, 穆维新

(郑州大学 信息工程学院, 河南 郑州 450001)

**摘 要:** 无线传感器网络中网络拓扑对配置网络资源、优化网络性能具有十分重要的作用. 根据在汇聚节点收集到的网络节点报文接收/丢失的信息, 提出了一种基于数据聚合的分层拓扑发现算法, 该算法同时可以实现对传感器网络节点的分层. 针对目前的拓扑发现算法在网络链路丢包率较低的情况下无法快速推断出网络拓扑的问题, 提出了一种使部分节点进入睡眠状态的方法推断低丢包率条件下的网络拓扑. 仿真结果表明, 提出的算法可以利用较少的观测数据准确快速地推断传感器网络的拓扑.

**关键词:** 无线传感器网络; 网络层析成像; 拓扑推断; 数据聚合

**中图分类号:** TP393

**文献标志码:** A

### 0 引 言

无线传感器网络 (Wireless Sensor Network, WSN) 在民用和军事领域具有广泛的应用, 为了更好地监测、维护无线传感器网络, 优化网络性能, 需要实时地获取多种网络状态参数. 网络拓扑作为传感器网络重要参数在网络的监测、维护、资源分配等工作中起着重要作用.

由于无线传感器网络中节点的能量、通信资源和计算资源十分有限<sup>[1]</sup>, 测量网络参数时应尽量不增加网络负担. 网络层析成像技术通过网络边缘节点的测量数据, 可以在不增加网络负担的前提下推断出网络内部的相关性能参数<sup>[2-4]</sup>. 目前关于传感器网络层析成像技术的研究主要集中在报文丢失率测量和能量测量上, 而且很多都是基于传感器网络逻辑拓扑已知的前提下进行的<sup>[5-6]</sup>, 涉及传感器网络逻辑拓扑获取方法的文献较少. ZHAO Tao 等人提出了基于数据融合的拓扑识别算法<sup>[7-8]</sup>. 其中文献[8]中的 HHC 算法只使用在 Sink 处得到的观测数据, 通过比较相邻层节点间报文到达序列的 Hamming 距离来推断网络拓扑, 该算法不需内部节点协作, 但需要事先知道各节点到 Sink 的跳数<sup>[7]</sup>. 文献[7]中的 MPIDA 方法利用节点及其祖先节点的报文丢失关系, 判断出节点的祖先集, 进而推断出节点的父子关系, 再逐条路径地重构出网络拓扑. 这种方法在观测数据较少或者网络规模较大时, 容易产生误

判<sup>[6]</sup>, 且其推断速率较慢.

笔者在 MPIDA 方法的基础上利用传感器网络节点的分层信息, 提出一种基于数据聚合的分层拓扑识别算法 (Level-Topology Identifying Algorithm, LTIA). 该算法利用端到端的报文到达/丢失信息, 推断每个节点的祖先节点和各个节点所在的层, 根据节点所在的层推断出网络中节点的父子关系, 从而重构整个网络拓扑.

### 1 系统模型

传感器网络拓扑指的是传感器网络中路由的逻辑关系, 并假定网络拓扑静态不变, 这是可行的<sup>[9]</sup>. 笔者提出的拓扑推断算法只需要网络拓扑在 Sink 节点收集到足够的报文到达/丢失信息前保持不变即可.

笔者采用文献[5]提出的网络模型, 其中拓扑模型为数据聚合形成的反向多播树 (图 1). 模型为用  $T = (V, L)$  来表示数据融合形成的反向多播树.  $V$  是节点集, 包含 Sink 和一般传感器, 每个节点都有唯一的标识. 约定用  $s$  表示 Sink, 它是  $T$  的根节点. 其他节点发送并汇聚感知数据到  $s$ .  $L$  是链路集,  $L$  的元素  $(j, i) \in V \times V$  对应网络中节点  $j$  到节点  $i$  的链路, 记为链路  $j, i$  称为  $j$  的父节点, 记为  $i = f(i)$ . 各条链路的报文丢失过程相互独立的 Bernoulli 过程. 链路  $i$  的成功传输概率记为  $a_i$ , 丢包率则记为  $\bar{a}_i = 1 - a_i$ . 用随机过程  $Z = (z_{i,j})$ ,  $i \in V \setminus \{s\}$ ,  $j \in a(i)$  描述流经反向多播树的数据流,

收稿日期: 2010-12-01; 修订日期: 2011-03-20

作者简介: 申金媛 (1966-), 女, 山西平遥人, 郑州大学教授, 主要研究方向为光电信号处理、图像识别, E-mail: jyshen@

zzu.edu.cn.

其中  $z_{i,j} \in \{0,1\}$ .  $z_{i,j}=1$  意味着节点  $i$  发出的数据成功到达节点  $j$ , 反之  $z_{i,j}=0$  表示数据在路径  $p_{i,j}$  上丢失. 记  $N$  为 Sink 收集到数据的轮数, 第  $m$  轮的输出为  $X^{(m)} = x_i^{(m)}, i \in V$ . 对每个节点  $i$ , 在 Sink 处能得到  $N$  个观测结果, 这些观测结果构成一个 0,1 序列:  $X_i = (x_i^{(m)}), i \in V, 1 \leq m \leq N$  约定  $\forall m: x_i^{(m)} = 1, x_i^{(m)}$  反映  $i$  的第  $m$  轮数据是否到达 Sink.

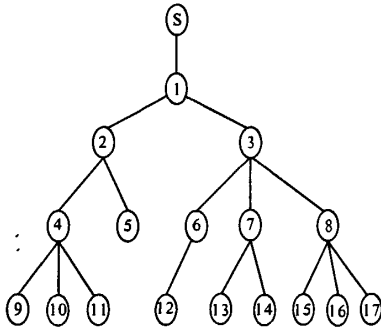


图1 网络拓扑示意图

Fig.1 Network topology

因为传感器网络中节点的带宽和能量有限,所以在收集数据过程中常采用数据聚合的方法.在数据聚合过程中,传感器网络中的节点通过一个倒的多播聚合树把收集到的数据转发送给汇聚节点.等收集到所有子节点转发过来的数据或定时器超时,该节点将聚合收到的所有子节点以及本身的数据转发给它的下一跳节点.

## 2 分层拓扑识别算法(LTIA)

若能找到反向多播树中每个节点所在的层数及其父节点,就能构建出整个树状拓扑.

### 2.1 推断节点的祖先集

根据在汇聚节点收集到的网络节点报文接收/丢失的信息,逐一判断节点  $k \neq s, k \in V$  的祖先节点集  $F(k)$ ,先假设  $F(k) = V$ .在数据聚合的过程中如果一个节点丢失了它的聚合数据,那么它所有的子孙节点的数据都会丢失而不能到达汇聚节点,即如果  $x_i^{(m)} = 0$ ,那么  $x_k^{(m)} = 0, \forall k \in d(i)$ .换言之,在  $m$  轮数据收集过程中如果  $x_i^{(m)} = 0$ ,而  $x_k^{(m)} = 1$ ,则说明  $i$  不可能是  $k$  的祖先节点.根据这一数据丢失关系逐一排除  $F(k)$  中的非祖先节点,即可以得到节点  $k$  的祖先节点集  $F$ .

### 2.2 实现传感器网络的分层

在得到网络中节点的祖先集  $F$  的情况下,具有相同祖先节点数的节点即为同一层的节点.因

此通过网络中所有节点的祖先集可以实现对网络中节点的分层.

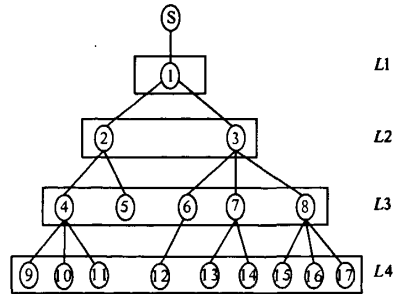


图2 网络节点分层拓扑图

Fig.2 Network level-topology

### 2.3 确定节点的父节点

根据上一步骤获得的传感器网络的分层可以知道每一个节点所在的层数,那么在节点  $k$  的祖先集  $F$  中与  $k$  节点层数相邻的节点即为  $k$  节点的父节点.在得到每一个节点的父节点之后,从 Sink 节点开始,就可以逐层的重构出网络拓扑  $T$ .

### 2.4 算法具体实现

(1)初始化:输入节点集  $V$ ,以及在汇聚节点  $s$  处收集到的各节点的数据集  $X$ ,数据收集轮数  $N$ ;对于每一个节点  $k \neq s, k \in V$ ,其祖先节点集  $F(k) = V$ ;

(2)当  $x_k^{(m)} = 1$  时,对每一个节点  $i \neq s$ ,如果有  $x_i^{(m)} = 0$ ,那么将  $i$  从节点  $k$  的祖先集中剔除  $F(k) = F(k)/i$ ;

(3) $F(k)$ 中元素的个数即为节点  $k$  所在的层  $r$ ;节点  $k$  的父节点即为  $F(k)$  中层数为  $r-1$  的节点;

(4)从 Sink 节点开始,就可以逐层的重构出网络拓扑  $T$ .

## 3 部分节点休眠的分层拓扑发现算法

一些实时监控的传感器网络对数据完整性要求较高,网络链路丢包率较低,还有一些传感器网络节点使用有源电源供电如 ZigBee 网络,节点死亡率低.利用目前的拓扑推断算法推断这类传感器网络的拓扑结构需要更多的观测数据,因为这些算法是基于网络中的报文丢失信息来推断的,当链路丢包率较小时,在 Sink 节点处获得的报文端到端的丢失信息比较少.为了在网络性能较好的情况下较快的推断出网络拓扑,笔者提出一种使部分节点休眠的分层拓扑发现算法(Node

Sleeping Level-Topology Identifying Algorithm, NSLTIA). NSLTIA 算法的主要思路是通过 Sink 节点向网络中的其它节点按一定的比例随机发送休眠消息,使网络中的部分节点进入休眠状态,从而在下一轮的数据收集过程中这些节点不再进行数据的收集和转发,这在传感器网络中是可行的<sup>[1]</sup>. NSLTIA 方法在拓扑推断过程中虚拟增加了网络中链路的丢包率,在汇聚节点可以更快的获得网络节点报文丢失信息,从而提高算法推断出拓扑的速度.

## 4 性能仿真及比较

### 4.1 LTIA 与 MPIDA 仿真对比

笔者采用 Matlab 软件对 LTIA 算法与 MPIDA 算法进行仿真对比. 仿真在两种规模的传感器网络下进行:一种是在图 1 所示的 18 个节点的传感器网络,其拓扑结构为倒多播树,其中  $s$  为 Sink 节点,它通过无线链路收集其它节点的数据;一种是由 GT-ITM 软件产生的 120 个节点的传感器网络. 在每一轮数据采集的过程中,一个节点向其父亲节点发送报文,是否发送成功是随机的,但是每条链路都有一个预定的报文丢失率. 仿真在丢包率分别为 1% ~ 10% 和 15% ~ 20% 两种情况下对算法进行仿真. 定义拓扑相似度为正确识别出的节点个数与总节点个数的比值. 被正确识别节点指的是该节点所在的层和父节点都被正确的识别出来.

图 3 图 4 分别表示在链路丢包率为 1% ~ 10% 时不同网络规模下 MPIDA 算法和 LTIA 算法的拓扑相似度与数据收集轮次之间的关系. 其中图 3 所示对于 18 个节点的网络,要使网络拓扑推断正确率达到 95%, MPIDA 算法需要 50 轮观测数据,而 LTIA 算法仅需 35 次. 图 4 表示在 120 个节点的网络中,LTIA 方法需 480 轮观测数据即可使推断正确率达到 100%,而 MPIDA 方法需要 800 轮观测数据. 图 5、图 6 是链路丢包率为 15% ~ 20% 时不同网络规模情况下的仿真结果. 图 5 中,MPIDA 算法需要 60 次数据收集才能正确的识别出 18 个网络节点,LTIA 算法仅需 30 轮观测数据就能全部把内 18 个网络的节点正确识别出来. 通过图 6 看出,对于 120 个节点的传感器网络 MPIDA 算法需要 350 轮数据收集才能使准备率达到 95%,而 LTIA 算法仅需 260 轮数据收集即可推断出网络拓扑.

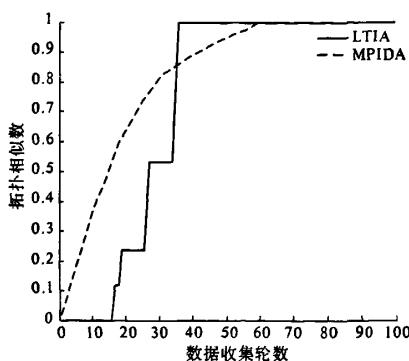


图 3 丢包率为 1% ~ 10% 时 18 个节点传感器网络拓扑相似度与数据收集轮数的关系

Fig. 3 The relationship between topology similarity and data collection rounds in 18 nodes WSN when loss rate is 1% ~ 10%

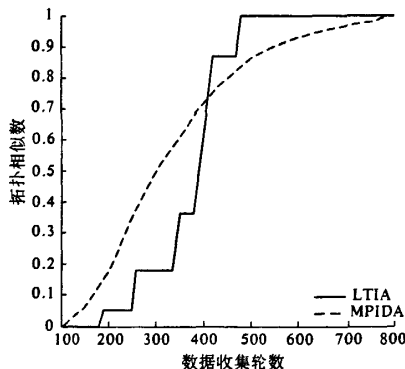


图 4 丢包率为 1 ~ 10% 时 120 个节点传感器网络拓扑相似度与数据收集轮数的关系

Fig. 4 The relationship between topology similarity and data collection rounds in 120 nodes WSN when loss rate is 1% ~ 10%

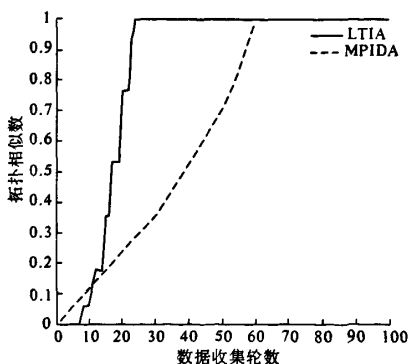


图 5 丢包率为 15% ~ 20% 时 18 个节点传感器网络拓扑相似度与数据收集轮数的关系

Fig. 5 The relationship between topology similarity and data collection rounds when loss rate is 15% ~ 20% in 18 nodes WSN

对比 MPIDA 算法,笔者提出的 LTIA 算法正确推断出网络拓扑所需的数据收集次数要少,这是因为 LTIA 算法利用了网络节点的分层信息,而不用逐条链路进行拓扑推断.从仿真结果中可以看出在算法推断初期,LTIA 算法的拓扑相似度不如 MPIDA 算法,这是因为 LTIA 算法需要一定数量的收集数据才能获得网络节点的分层信息.

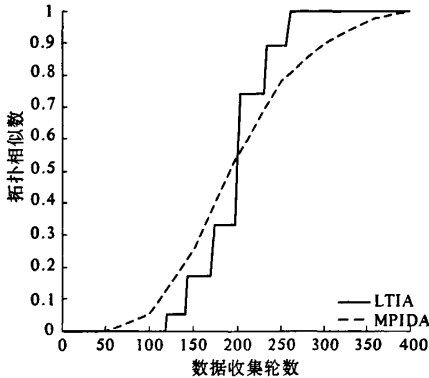


图6 丢包率为15%~20%时120个节点传感器网络拓扑相似度与数据收集轮数的关系

Fig.6 The relationship between topology similarity and data collection rounds in 120 nodes WSN when loss rate is 15% ~ 20%

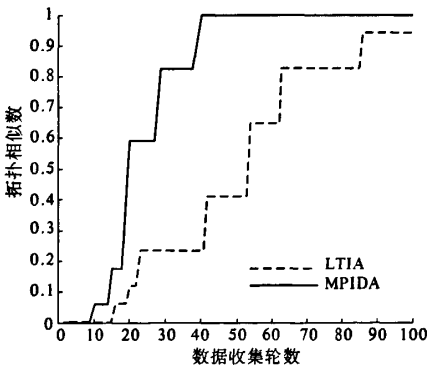


图7 部分节点进入休眠状态后18个节点传感器网络拓扑相似度与数据收集轮数的关系

Fig.7 The relationship between topology similarity and data collection rounds in 18 nodes WSN when some nodes were sleeping

#### 4.2 NSLTIA 与 LTIA 仿真对比

从图7、图8可以看出,在网络链路丢包率为1%~5%的情况下在18个节点的传感器网络中要使网络拓扑推断正确率达到95%需要85轮观测数据,部分节点进入节能状态后,只需要40轮观测数据即可使网络拓扑推断正确率达到95%.而在120个节点的传感器网络中,NSLTIA方法使

网络拓扑推断正确率达到95%所需的数据观测轮数可以节省一半以上.即使在链路丢包率较小的情况下NSLTIA算法也可以用较少的观测数据准确的推断出网络拓扑.

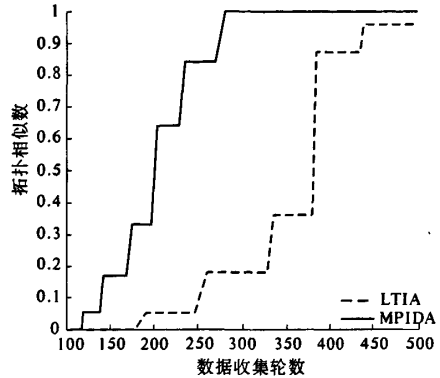


图8 部分节点进入休眠状态后120个节点传感器网络拓扑相似度与数据收集轮数的关系

Fig.8 The relationship between topology similarity and data collection rounds in 120 nodes WSN when some nodes were sleeping

#### 5 结论

笔者提出的算法能在无需中间节点协作、不增加网络负载的情况下,利用较少的观测数据准确地推断出不同环境下的网络拓扑.当网络规模扩大、节点数量增加时,具有相同报文到达/丢失特性的节点数量也随之增加,从而需要更多的观测数据来判断这类节点的祖先节点和父节点,此时需要更多的观测轮数来推断网络拓扑结构.

#### 参考文献:

- [1] LAN F A, WEILIAN S, YOGESH S, et al. A survey on sensor networks[J]. IEEE Communications Magazine, 2002, 40(8): 102 ~ 114.
- [2] BU T, DUFFIELD N, PRESTI F L, et al. Network tomography on general topologies[J]. ACM SIGMETRICS Performance Evaluation Review, 2002, 30(1): 21 ~ 30.
- [3] MARK C, ALFRED H, ROBERT N, et al. Internet Tomography[J]. IEEE Signal Processing Magazine, 2002, 19(3): 47 ~ 65.
- [4] CASTRO R, MARK C, LIANG G, et al. Network tomography: Recent developments[J]. Statistical Science, 2004, 19(3): 499 ~ 517

(下转第118页)

-450.

- [6] WANG Peng-tao, JIANG Ren-qiu, SONG Fu-yuan, et al. Numerical simulation and analyze of three-dimensional strong swirl flow in the vortex tube[J]. Cryogenics & Superconductivity, 2007, 35(1): 73 - 78, 83.

## Analysis of Flow and Heat Transfer in High-Pressure Vortex Tube for LNG

KONG Song-tao, LIU Juan, ZHAO Li-jun, CAI Ping, HUANG Yue

(School of Mechanical Engineering Chongqing University of Science and Technology, Chongqing 401331, China)

**Abstract:** The high-pressure vortex tube is a kind of key equipment in skid-mounted package liquefied CBM/NG. As a strong centrifugal force, the three-dimensional measurement to intense rotating flow is very different from the real situation. The method of numerical computing is used to study the high-pressure vortex tube flow field and temperature field. The pressure difference between hot and cold side and the special nozzle are conducive to the formation of supersonic flow, and lead to lower temperature in cold-side. The study also shows that to control the back pressure of hot end in high-pressure vortex tube will help to achieve higher cooling efficiency.

**Key words:** vortex tube; LNG; skid-mounted

(上接第 114 页)

- [5] HARTL G, LI Bao-chun. Loss inference in wireless sensor networks based on data aggregation[C]//In proceedings of the Third IEEE/ACM International Symposium on Information Processing in Sensor Network. New York: ACM Press, 2004:396 - 404
- [6] 张志勇, 胡光岷. 无线传感器网络拓扑识别算法[J]. 计算机应用, 2010, 30(3): 733 - 736
- [7] ZHAO Tao, CAI Wan-dong, LI Yong-jun. MPIDA: A sensor network topology inference algorithm[C]//2007 International Conference on Computational Intelligence and Security. Washington DC: IEEE Computer Society, 2007: 451 - 455.
- [8] ZHAO Tao, CAI Wan-dong, LI Yong-jun. Using end to end data to infer sensor network topology[C]//2007 IEEE International symposium on signal processing and Information Technology. Washington DC: IEEE Computer Society, 2007: 504 - 508.
- [9] 陈荣, 江虹, 杨欢. Ad Hoc 网络路由协议性能对比仿真研究[J]. 计算机工程与设计, 2008, 29(18): 4672 - 4675.

## A Level-Topology Identifying Algorithm in Sensor Network

SHEN Jin-yuan, ZHAO Xu-dong, LIU Run-jie, MU Wei-xin

(1. School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China)

**Abstract:** Sensor network topology plays an important role in resource management and deployment. By using the lose/receive information of data collected in the sink, we proposed a level-topology identification algorithm based on data aggregation in this paper, and we can get the level information of the sensor network in the same time. Considering the low speed of all topology algorithms proposed when the loss rate of sensor network is low, we proposed a method which put some nodes into sleeping state to inferring the topology under low loss rate. The simulation shows that the proposed algorithm can infer the sensor network topology accurately and quickly.

**Key words:** sensor network; network tomography; topology inference; data aggregation