

# 基于节点感应度的自主决策目标跟踪算法

郑 瑾 吕朋朋 吕邵明

(中南大学信息科学与工程学院 长沙 410083)

**摘 要** 目标跟踪是无线传感器网络的一项基本应用。由于传感器节点能量有限,如何在保证跟踪精度的前提下降低节点能耗以延长网络生命周期一直是研究的重点之一。文中采用 RNG 平面化技术将无线传感器网络平面化为多边形跟踪结构,基于定位边,采用加权质心算法对目标位置进行估计,结合节点对目标感应质量与距离负相关的特性,给出了节点感应度的计算方法,同时提出了一种节点自主决策是否参与跟踪的目标跟踪算法(NS-ADTT)。该算法中,节点可根据自身感应度值及局部网络情况自主决策是否参与当前跟踪。仿真结果表明,在目标跟踪过程中,该算法在保证一定跟踪精度的基础上减少了参与跟踪的节点数,降低了网络能耗,有效地延长了网络生命周期。

**关键词** 无线传感器网络,目标定位,目标跟踪,节点感应度,节点邻近域,能量有效

**中图分类号** TP393 **文献标识码** A

## Node Sensitivity-based Autonomous Decision-making Target Tracking Algorithm

ZHENG Jin LV Peng-peng GUO Shao-ming

(School of Information Science and Engineering, Central South University, Changsha 410083, China)

**Abstract** Target tracking is one of the basic applications in wireless sensor networks (WSNs). Because of the limitation of node energy, how to reduce node energy consumption to ensure the tracking accuracy is the major research. In this paper, the wireless sensor networks were divided into several polygons using RNG planarization technology, and weighted centroid algorithm based on the polygon structure was used to estimate the target's position. Since the node sensor quality is inversely proportional to the distance, we gave the calculation method of node sensitivity, and proposed a node sensitivity-based autonomous decision-making target tracking algorithm (named NS-ADTT). According to the node sensitivity and the local network, the node can make decision to participate or not in the current tracking. The simulation results show that our algorithm can reduce the tracking number of nodes and the energy consumption of network, and extend the network lifetime efficiently.

**Keywords** Sensor networks, Target locating, Target tracking, Node sensitivity, Node adjacent domain, Energy efficiency

### 1 引言

无线传感器网络(WSNs)由大量具备数据处理和无线通信能力的微型传感器节点组成,是一种面向任务的无线自组织网络系统<sup>[1]</sup>。

目标跟踪是无线传感器网络最具吸引力和代表性的基础应用之一,可以被单独应用,也可以作为物联网应用的组成部分。例如,近年来出现了集成 RFID 功能的传感器,它对目标跟踪的广泛应用起着较大的推动作用。但是,由于传感器节点自身能量的限制以及周围环境的影响,使得目标跟踪技术要从研究走向应用还面临着很多挑战。

基于结构的目标跟踪算法,是指通过一定的结构将跟踪节点进行组织,跟踪节点集合中有一个节点收集、融合跟踪信息,并向 Sink 节点或用户返回跟踪结果。其中,基于“面”结构的无线传感器网络中每个节点的相邻节点集合可知,通过合理调整节点发射功率,可以建立一个连通的网络,同时尽可

能降低网络能耗。基于此,研究者提出了一些基于“面”的目标跟踪协议<sup>[2,4,6]</sup>。

文献[2]基于面路由和信息驱动,提出了监测与备份机制的目标跟踪算法(TTMB),该算法利用 GG 算法<sup>[3]</sup>将监测区域划分为多个“面”,在与目标处于同一“面”的节点中,距离目标最近的节点被选择为跟踪节点,跟踪节点在其近邻中选择一个节点作为跟踪信息的备份节点。在后续跟踪过程中,利用线性预测模型预测目标将要到达的位置,并据此选择下一时刻的跟踪节点与备份节点。该算法采用高斯拟合的线性预测机制提高目标位置预测精度。同时,为了提高系统容错性,在跟踪节点失效时由备份节点来担任跟踪节点。但是,该算法在目标移动随机性较强的场合下跟踪误差较大;在每个跟踪时刻,跟踪节点需要与相邻节点通信来选择备份节点,带来额外通信能耗。

文献[4]结合面路由及节点序列提出了一种能量高效的目标跟踪算法(NSFS),该算法基于“面”结构的网络模型,用

到稿日期:2013-08-22 返修日期:2013-10-25 本文受国家自然科学基金(61272496)资助。

郑 瑾(1970—),女,博士,副教授,主要研究方向为移动计算、无线传感器网络、软件工程等,E-mail:zhengjin@csu.edu.cn;吕朋朋(1987—),男,硕士生,主要研究方向为无线传感器网络等;吕邵明(1989—),男,硕士生,主要研究方向为无线传感器网络等。

传感器节点间的中垂线对网络再次划分,并用节点序列表示二次划分后的子区域。在移动目标进入监测区域后,根据节点监测到目标的时间先后,形成一个跟踪过程中的节点序列,基站采用 EKT 算法<sup>[5]</sup>匹配两个节点序列,根据匹配结果估计目标所在区域,并将该区域的质心作为目标的估计位置。基于此,该算法采用线性预测的机制唤醒目标将要到达的“面”内节点参与跟踪。该算法采用节点序列估计目标位置,有效减少了节点间数据通信量,且结合预测机制,减少了参与跟踪节点的数目,有效降低了网络能耗。但是,该算法随着网络中节点数目的增多,匹配节点序列算法复杂度会显著提高,不适用于大规模的传感器网络,且采用线性预测机制导致目标跟踪精度较低。

文献[6]在面路由算法的基础上提出了一种基于多边形结构的目标跟踪算法(PTT)。该算法利用 GG 算法将传感器网络平面化为多边形结构,目标所处的多边形称为“活动多边形”,活动多边形内的节点都处于活动状态以参与跟踪,目标将要到达的多边形称为前向多边形,相邻两个多边形的公共边称为关键边。该算法用目标所经历过的活动多边形序列以及所选择的关键边序列来对目标的运动轨迹进行刻画。在实施目标跟踪的过程中,关键边两端的节点负责对目标进行跟踪,并根据监测到的目标运动信息及距离关键边的距离来通知目标运动方向上的多边形内的节点进入唤醒状态,一旦目标跨过关键边进入前向多边形,该多边形就成为新的活动多边形,根据目标的运动情况选择新的关键边。同时,为了延长网络生命周期,通知先前活动多边形内的节点进入睡眠状态。该算法目标丢失率较低,且能够完成实时跟踪任务。但是,它采用多边形序列刻画目标移动轨迹,没有考虑了目标在多边形内的定位与跟踪,跟踪精度较粗糙,且在跟踪过程中需要目标所处多边形中的节点都处于跟踪状态,参与跟踪的节点数太多,增大了网络节点的能量消耗。

本文充分考虑了节点局部拓扑以及剩余能量;在对目标位置进行估计之后节点自主决策是否参与跟踪;减少参与跟踪的节点数,同时降低跟踪过程中节点之间的通信负载。

## 2 网络模型及相关定义

### 2.1 网络模型

假设二维平面下所部署的传感器网络满足以下基本条件:

(1)网络是同构的,所有传感器节点拥有相同的感知能力、通信能力、处理能力和初始能量。

(2)节点被合理部署在监测区域。所部署的传感器网络具有良好的覆盖性,采用 RNG<sup>[7]</sup>平面化处理呈封闭的多边形结构,且不会出现文献[8]所讨论的覆盖漏洞。

(3)每个传感器节点通过自定位(self-localization)算法<sup>[9]</sup>获知自己的位置。在传感器网络部署完成后,节点就开始启动自定位过程,在后续的目标跟踪中,传感器节点的位置都是已知的。

(4)当一个传感器节点检测到目标时,它可以根据接收到目标信号的载波功率估算出自己与目标之间的距离。

### 2.2 相关定义

#### 2.2.1 节点感应度及定位边

在传感器节点对目标的识别过程中,目标距离节点越近,

节点的识别能力越强,超出一定距离范围,节点将无法识别目标。本文结合文献[10]中讨论的传感器节点的感知模型,用节点感应度的概念描述节点对目标的识别能力与其距离负相关的特性。

**定义 1(节点感应度)** 记  $t$  时刻节点  $S_j$  监测到与目标的距离为  $d_j(t)$ ,在不大于感应半径  $r$  时,节点  $S_j$  的节点感应度为 1;当  $d_j(t)$  大于一定距离阈值  $\sigma_a$  时,本文设定为通信半径  $R$ (假设通信半径  $R$  为感应半径  $r$  的 2 倍,即  $2r$ ),感应度为 0;当  $d_j(t)$  在  $(r, \sigma_a)$  区间时,感应度为  $(0, 1)$  区间的值,该值与当前节点感应到的与目标之间的距离负相关。其定义式如式(1)所示。

$$S(d_j(t)) = \begin{cases} 0, & d_j(t) \geq \sigma_a \\ \frac{\sigma_a - d_j(t)}{\sigma_a - r}, & r < d_j(t) < \sigma_a \\ 1, & d_j(t) \leq r \end{cases} \quad (1)$$

**定义 2(定位边)** 在多边形结构中,距目标最近的两个相邻节点之间的边称为定位边。当目标进入节点监测区域内时,感应度不为 0 的相邻节点相互通信,距离目标最近的两个相邻节点形成定位边。其中,定位边中距离目标最近的节点称为定位边的主端点,若定位边两端节点距离目标等距离,则在目标运动方向上的节点被确定为定位边的主端点。在确定定位边之后,主端点与相邻非定位边中的相邻节点通信,确定目标所处多边形以及定位节点集  $LocS$ 。某一时刻定位边具体的选择过程如下所示。

**步骤 1** 节点感应度不为 0 的  $S_i$  与其相邻节点通信,若存在比自己距离目标近的节点,则退出定位边选择。否则,转到步骤 2。

**步骤 2**  $S_i$  对目标位置进行估计,并预测所需能量,若自身剩余能量不充足,则置自身感应度值为 0,转到步骤 1。否则,转到步骤 3。

**步骤 3** 在  $S_i$  相邻节点中选择距离目标最近的节点,记为  $S_j$ ,若  $S_j$  距目标的距离大于  $S_i$  与  $S_j$  之间的边长,则转到步骤 1。否则,得到定位边为  $LEs = \{S_i, S_j\}$ ,其中定位边中主端点为  $S_i$ ,结束。

通过上述方法得到定位边之后,选择与主端点  $S_i$  相邻的节点中除了  $S_j$  之外距离目标最近的节点,记为  $S_k$ ,至此参与定位的节点集  $LocS = \{S_i, S_j, S_k\}$ ,然后结合文献[11]对基于无线传感器网络拓扑结构的定位模型的分析,采用文献[12]提出的加权三边定位算法对目标位置进行估计。

假设当目标处于多边形  $P$  内,该多边形中不存在定位边,即多边形  $P$  内存在一点  $o$  满足  $\forall S_{i-1}, S_i, S_{i+1} \subseteq S, |oS_i| \geq |S_{i-1}S_i|, |oS_i| \geq |S_iS_{i+1}|$ ,分别以各节点为圆心,以该点所在边为半径画圆,若点  $o$  不存在于任一个圆内,则称多边形内的这类点(或点集)为多边形的覆盖漏洞。其中  $S = \{S_1, S_2, \dots, S_i, \dots, S_n\}$  是多边形  $P$  中的节点集合。本文不考虑存在多边形覆盖漏洞的传感器网络部署,即当目标处于某个多边形内时,该多边形必定存在定位边。因此,根据定位边选择原理,当目标处于多边形内时,一定存在定位边。

#### 2.2.2 节点邻近域

采用 RNG 算法对传感器网络进行平面化处理,这样就消除了传感器网络中交叉边的存在,将网络划分为简单多边形。

形结构。如图 1 所示,某一部分监测区域的传感器网络被划分为多边形结构  $P_0, P_1, P_2, P_3, P_4$ 。

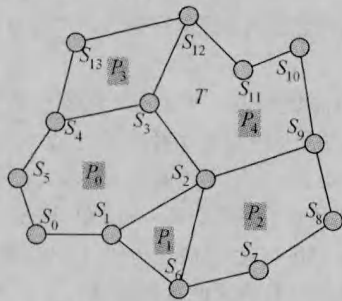


图 1 RNG 算法平面化的多边形传感器网络

考虑到传感器网络中节点部署的随机性,平面化得到的简单多边形中,若多边形中某内角小于  $180^\circ$ ,则称其为多边形的凸顶点,否则,称其为多边形的凹顶点。

**定义 3(节点邻近域)** 将凸顶点的邻近域定义为该顶点所在的两条边与其垂直平分线所围成的四边形;对于凹顶点的邻近域,将以该节点所在的两条边中较大边的一半为半径做的圆与当前多边形围成的区域定义为该顶点的邻近域。如图 2 所示,凸顶点  $S_9$  和  $S_{10}$  的邻近域分别为  $S_{nad}^9$  和  $S_{nad}^{10}$ ;凹顶点  $S_{11}$  的邻近域为  $S_{nad}^{11}$ 。

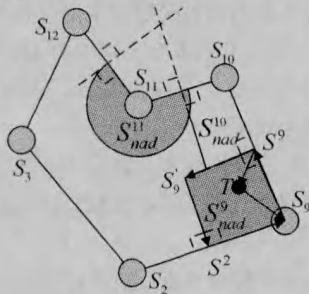


图 2 多边形顶点邻近域

在传感器网络经过平面化处理之后,每个节点能够获得相邻节点信息以及所属的多边形结构,然后根据定义 3 能够获得自身的邻近域信息,并参考文献[13]中提出的多边形方向的判断方法获得节点邻近域的方向,基于此,结合目标的估计位置判断目标是否存在于节点邻近域。如图 2 所示,当目标  $T$  在监测区域内时,定位边为  $LEs = \{S_9, S_{10}\}$ ,定位节点集  $LocS = \{S_2, S_9, S_{10}\}$ ,节点  $S_9$  根据节点定位集采用加权质心算法对目标位置进行估计,然后判断  $\triangle S_9 S^9 T$  与其节点邻近域  $S_9 S^9 S_9' S^2$  方向是否一致,若方向相同,则说明目标处于自身邻近域内,否则,认为目标没有处于自身邻近域内。

### 3 算法描述

目标跟踪过程包括 3 个阶段:检测、定位和通告。根据多边形结构,由定位边、定位节点集  $LocS$ ,采用加权质心定位算法对目标位置进行估计。本文假设传感器节点有 4 个基本状态,在获得目标估计位置后,基于四状态转换机制提出了一种节点自主决策是否参与目标跟踪的算法(NS-ADTT)。

#### 3.1 节点状态定义及转换机制

设传感器节点有睡眠、检测、定位和跟踪 4 个状态。具体定义如下:

睡眠状态:关闭节点所有功能模块。

检测状态:打开感应功能模块和处理器功能模块,节点根

据监测到的目标信息计算节点感应度,通信功能模块仍然关闭。

定位状态:打开所有功能模块,对目标进行感应、数据处理以及与相邻节点进行通信,从而对目标位置进行估计,是决定节点是否参与目标跟踪的过渡状态。

跟踪状态:打开所有功能模块,节点主要对目标的实时数据进行感应、处理,完成目标位置估计,可以与通信半径范围内的节点甚至 Sink 节点进行通信,只有跟踪状态下的节点才有可能参与跟踪。

网络中所有传感器节点都周期性地处于睡眠和检测状态,基于节点感应度及定位边定义了上述节点 4 个状态进行转换的条件:

- ①睡眠周期结束;
- ②检测周期结束;
- ③节点感应度(Node Sensitivity, NS)为 1,即  $NS=1$ ;
- ④  $0 < NS < 1$ ;
- ⑤定位边中主端点;
- ⑥非定位边中主端点;
- ⑦目标在节点邻近域内;
- ⑧目标不在节点邻近域内;
- ⑨目标丢失;
- ⑩剩余能量不足。

状态的具体转换过程如图 3 所示。

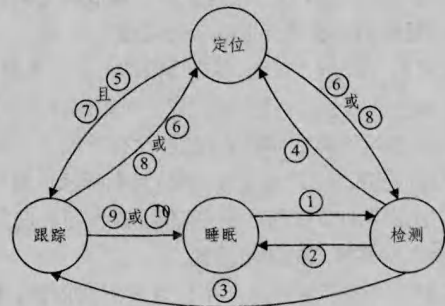


图 3 传感器节点状态转换图

跟踪过程中状态的具体转换过程如下:

步骤 1 目标进入无线传感器网络监测区域,节点计算感应度值,若  $NS=1$ ,则转到步骤 2;若  $0 < NS < 1$ ,转到步骤 3;若  $NS=0$ ,则检测状态等待周期结束后进入“睡眠”状态,结束。

步骤 2 节点转换为“跟踪”状态,估计参与本次跟踪所消耗的能量,若剩余能量充足,则转到步骤 4,否则,节点转到“睡眠”状态,结束。

步骤 3 节点转换为“定位”状态,如果相邻节点中没有“跟踪”状态的节点,则转到步骤 4,否则,转到“睡眠”状态,结束。

步骤 4 对于“跟踪”状态下的节点,首先,判断当前节点是否为定位边中主端点,若是,则对目标位置进行估计,从而判定目标是否在该节点邻近域,如果同时满足上述两个条件,该节点被选择为跟踪节点,否则将转变为“定位”状态。

特别地,为了降低目标丢失率,若节点感应度不为 0,节点将继续保持检测状态而不进行睡眠状态的转换。

#### 3.2 目标跟踪算法描述

网络中节点周期性地处于检测和睡眠状态,处于检测状

态的节点对目标进行感应,感应度不为0的节点进行状态转换从而确定定位边及定位节点集。目标周围节点利用本文提出的目标跟踪算法(NS-ADTT)自主决策参与跟踪。算法中涉及到的条件及定义如表1所列。

表1 定义描述

定义(Terms)	描述(Descriptions)
TL	目标当前的估计位置(Target Location, TL)
NS	节点当前感应度(Node Sensitivity, NS), 由式(1)得
LEs	定位边节点集(Locating Edge set, LEs), 我们将其中主端点集合用 LEms 表示
LocS	定位节点集(Locating Sensors set, LocS)
Ere	节点剩余能量(Residual Energy, Ere)
Eth	节点参与跟踪的能量阈值(Energy threshold, Eth)
NAD	节点邻近域(Node Adjacent Domain, NAD)
AdjNs	节点的相邻节点集(Adjacent Nodes set, AdjNs)
TSS	跟踪节点序列(Tracking Sensor Sequence, TSS)
TP	目标轨迹(Target Path, TP)

NS-ADTT 算法结合节点剩余能量、节点感应度、定位边及定位节点集,基于本文提出的四状态转换机制,选择跟踪节点。NS-ADTT 算法如图4所示。

```

算法1: NS-ADTT
input :Relative Neighborhood graph G
output:Tracking Sensors set TSS and
        Target Path TP
1 Initialization G and get sensors' NAD
2 LEs={}; LocS={}; TP={}; TSS={}
3 if sensor(i).NS > 0 then
4   Communicating with its AdjNs to get
   LEs, LEms and LocS
5 end if
6 Estimating target T's position (TL)
  with LocS and TP = TP ∪ {TL}
7 if LEms(i).Ere > Eth then
8   if TL ∈ sensor(i).NAD then
9     TSS=TSS ∪ {i}
10  end if
11 else Remove i from LEms
12   Set sensor(i).NS=0
13   if LEms == ∅ then
14     goto: 2
15   end if
16   i++ and goto: 6
17 end if
18 output TSS and TP
  
```

图4 跟踪算法 NS-ADTT

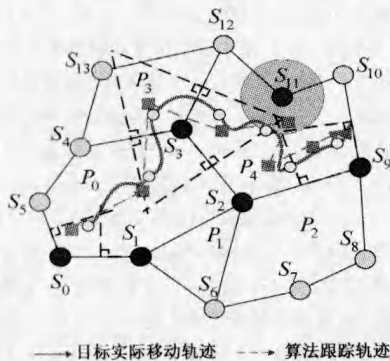


图5 跟踪移动目标示例

图5为基于NS-ADTT算法的跟踪实例,其中,定位边选择序列为  $S_0 S_5, S_1 S_2, S_3 S_4, S_3 S_2, S_{11} S_{10}, S_2 S_9, S_9 S_{10}$ ; 相应

地,跟踪节点序列为  $S_0 S_1 S_3 S_3 S_{11} S_4 S_9$ 。

## 4 模拟分析

本文在 Matlab 环境下对提出的 NS-ADTT 算法进行模拟,将与基于“面”结构的 TTMB<sup>[2]</sup>算法以及基于多边形结构的 PTT<sup>[5]</sup>算法从跟踪精度、网络能耗以及目标丢失率3个方面进行对比分析。有关节点在睡眠、检测以及通信过程中的能耗模型采用文献[14]提出的能量消耗模型,实验参数如表2所列。

表2 模拟参数表

参数	取值
区域大小	200m×200m
节点数量	100
通信半径(m)	40
感应半径(m)	20
目标移动速率(m/s)	0~15
目标加速度(m/s <sup>2</sup> )	[-5,5]
节点初始能量(J)	5
计算能耗(uJ)	15
感应能耗(uJ)	50
发送数据能耗(uJ)	750
接受数据能耗(uJ)	400
节点能量阈值(mJ)	1.5

图6示出在200×200m监测区域内,随机部署150个传感器节点,当目标穿过监测区域时,不同跟踪次数下NS-ADTT、PTT和TTMB算法跟踪精度的比较。在跟踪次数较小时,3个算法跟踪精度较低,随着跟踪次数的增加其跟踪精度得到提高。在跟踪过程中,NS-ADTT算法跟踪精度较高,PTT算法次之,TTMB算法相对较差。这是由于在采用NS-ADTT算法实施跟踪的过程中,用距离目标最近的3个节点采用加权质心定位算法对目标位置进行估计,在每个跟踪时刻得到的目标位置精度都相对较高;PTT算法中,用目标所在的多边形估计目标位置,存在不同时刻目标处于同一多边形情况,且当目标所处多边形结构较大时,跟踪误差较大;而TTMB算法则结合目标的历史信息采用线性预测机制估算目标下一时刻位置,由于目标运动的随机性,导致该算法平均跟踪误差较高。

图7示出跟踪次数为40,部署节点数量分别为25、50、100、150、200时NS-ADTT、PTT和TTMB算法跟踪精度的比较。由于这3个算法都基于平面多边形结构,在跟踪节点数目较少时跟踪精度相对较低,随着网络中节点数目的增加,跟踪精度呈提高的趋势,且NS-ADTT算法跟踪精度较好,PTT算法整体跟踪精度优于TTMB算法,TTMB算法最差。这是由于TTMB算法基于线性预测机制确定目标位置,从而选择跟踪节点,跟踪精度主要依赖于目标的运动情况以及所采用的预测机制,平均跟踪误差相对较大;PTT算法用多边形结构刻画目标位置,在网络中节点数目较少时跟踪误差较大,随着网络中节点密度的增大,跟踪精度得到提高,整体跟踪精度优于TTMB算法,且在节点数目在200时,跟踪精度逼近NS-ADTT算法;NS-ADTT算法考虑了每个跟踪时刻下目标所在多边形内部时的位置,且随着网络中节点数目的增加,采用质心算法估计的目标位置误差降低,其跟踪精度高于PTT算法。

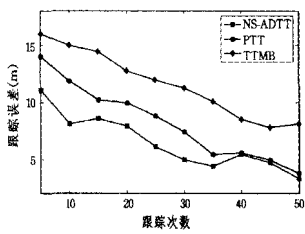


图6 不同跟踪次数下的跟踪误差

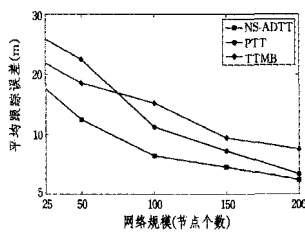


图7 不同节点数目下平均跟踪误差

如图8所示,跟踪次数为40、节点数量为100时,随着跟踪时间的增长,网络中无效节点数目增加,为了保证一定跟踪精度,需要对局部网络进行平面化处理从而消耗一定的能量,使得3种算法的网络能耗速度加快。其中,PTT算法需要目标所处多边形内所有节点处于活动状态,并进行通信来选择合适的节点参与跟踪,造成冗余节点以及冗余数据的存在;TTMB算法则结合面结构与预测机制选择跟踪节点,在一定程度上减少了跟踪过程中通信节点数目,但每个跟踪时刻都需要额外节点对跟踪节点进行备份。NS-ADTT算法中,节点根据目标的估计位置,定位边中主端点判断目标是否存在于自身邻近域内,从而无需与其他节点通信便可自主决策是否参与跟踪,有效地减少了参与跟踪的节点数与节点间的通信。因此,NS-ADTT算法网络能耗最低,TTMB次之,PTT算法网络能耗最高。

如图9所示,从算法的容错性考虑,PTT算法目标丢失率较低,TTMB算法次之,NS-ADTT算法目标丢失率较大,但其网络生命周期较长,整体目标丢失率在可承受范围内。这是由于,PTT算法要求目标所处多边形的节点都处于活动状态,关键边上的节点被选择为跟踪节点,在跟踪节点死亡或丢失目标时,活动多边形内的其他节点仍然可以监测到目标,从而降低了目标丢失情况;TTMB算法在选择跟踪节点时设置了相应的跟踪信息备份节点,提高了系统的容错性;本文提出的NS-ADTT在跟踪节点选择时充分考虑了节点剩余能量的影响,避免了由于能量耗尽造成网络黑洞而丢失目标的现象,但是,当跟踪目标的当前节点失效或者随着网络中失效节点数目的增加,造成多边形覆盖漏洞时,将会导致目标的丢失不可恢复。

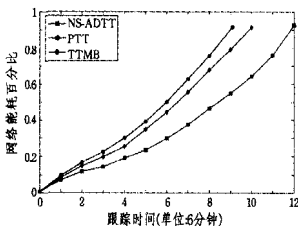


图8 网络能量消耗

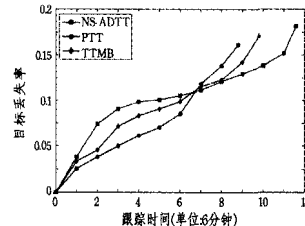


图9 目标丢失率

综上所述,在基于多边形的跟踪结构下,无线传感器网络实施跟踪的过程中,考虑跟踪精度与网络能耗,NS-ADTT算法在保证一定跟踪精度的基础上,有效地降低了节点的能量消耗,延长了网络生命周期。

**结束语** 基于多边形的跟踪结构,本文提出了一种节点自主决策是否参与跟踪的目标跟踪算法(NS-ADTT),并将之与其他采用同结构的跟踪算法进行了比较。采用该算法对目

标实施跟踪的过程中,能够感应到目标的节点与相邻节点进行通信,距离目标最近的3个节点采用加权质心算法对目标位置进行估计,在跟踪节点选择的过程中,由节点根据节点邻近域以及定位边的约束条件自主决策是否参与跟踪,减少了参与跟踪的节点数目以及节点之间的通信,从而在保证一定跟踪精度的基础上降低了网络能耗。

在对目标实施跟踪过程中,NS-ADTT算法的跟踪精度受网络拓扑结构的影响,对目标位置的估计决定了跟踪节点的选择。为了保证跟踪精度以及降低目标丢失率,未来工作的重点主要包括:研究一种基于多边形结构的目标定位机制;针对目标丢失的情况研究一种多边形结构下的恢复机制。

## 参考文献

- [1] 郑军,张宝贤. 无线传感器网络技术[M]. 北京:机械工业出版社,2012
- [2] Bhuiyan M Z A, Wang Guo-jun, Wu Jie. Target Tracking with Monitor and Backup Sensors in Wireless Sensor Networks[C]// Proceedings of the 18th IEEE International Conference on Computer Communications and Networks (ICCCN 2009). San Francisco, California, USA, August 2009
- [3] Gabriel K R, Sokal R R. A New Statistical Approach to Geographic Variation Analysis[J]. Systematic Zoology, 1969, 18: 259-278
- [4] 曾国定,李超良,王国军. 基于不可靠节点序列和面感知路由的目标跟踪[J]. 传感技术学报, 2011, 9(24): 1305-1310
- [5] Zhong Zi-guo, Zhu Ting, Wang Dan, et al. Tracking with Unreliable Node Sequences[C]// IEEE INFOCOM 2009. 2009: 1215-1223
- [6] Bhuiyan M Z A, Wang G, Wu J. Polygon-Based Tracking Framework in Surveillance Wireless Sensor Networks[C]// Proceedings of the 15th IEEE International Conference on Parallel and Distributed Systems (ICPADS 2009). Shenzhen, China, December 2009: 174-181
- [7] Toussaint G. The relative neighborhood graph of a finite planar set[J]. Pattern Recognition, 1980, 12(4): 261-268
- [8] 蒋杰,方力,等. 无线传感器网络最小连通覆盖集问题求解算法[J]. Journal of Software, 2006, 17(2): 175-184
- [9] Moses R L, Krishnamurthy D, Patterson R. A Self-Localization Method for Wireless Sensor Networks[J]. EURASIP Journal on Applied Signal Processing, 2003, 2003(4): 348-358
- [10] 柳立峰,邹仕洪,张雷,等. 基于概率覆盖模型的无线传感器网络密度控制算法[J]. 北京邮电大学学报, 2005, 28(4): 14-17
- [11] 顾晶晶,陈松灿,庄毅. 基于无线传感器网络拓扑结构的物联网定位模型[J]. 计算机学报, 2010, 33(9): 1548-1556
- [12] Liang Shao-hua, Bhuiyan M Z A, Wang Guo-jun. Auction-Based Adaptive Sensor Activation Algorithm for Target Tracking in WSNs[C]// Proceedings of the 2011 International Joint Conference of IEEE TrustCom-11/IEEE ICSS-11/FCST-11 (TrustCom 2011). Changsha, China, November 2011: 1217-1223
- [13] 李维诗,李江雄,等. 平面多边形方向及内外点判断的新方法[J]. 计算机辅助设计与图形学学报, 2000, 12(6): 405-407
- [14] Olariu S, Stojmenovic I. Design Guidelines for Maximizing Lifetime and Avoiding Energy Holes in Sensor Networks with Uniform Distribution and Uniform Reporting[C]// Proceedings of the 25th Conference on Computer Communications (INFOCOM). 2006: 1-12