

# 无线传感器网络自适应目标跟踪节点调度算法

卢旭<sup>1,2</sup>, 程良伦<sup>1</sup>, 罗世亮<sup>1</sup>

(1. 广东工业大学 自动化学院, 广东 广州 510006; 2. 广东技术师范学院 自动化学院, 广东 广州 510665)

**摘要:** 在无线传感器网络中, 设计合理的节点调度算法是提高网络感知能力、降低系统能耗的关键。在分析节点能耗模型的基础上, 针对移动目标跟踪型网络应用, 提出一种高能效的无线传感器网络自适应节点调度算法 ANSTT。该算法根据节点对移动目标的感知能力, 以及节点的相对剩余能量水平, 自动调整节点工作模式。仿真实验表明, ANSTT 算法在维持低感知延时、高目标感知率的同时, 可有效降低系统能耗, 延长网络寿命。

**关键词:** 无线传感器网络; 节点调度; 目标跟踪; 自适应; 高能效

**中图分类号:** TP393

**文献标识码:** A

## Adaptive node scheduling algorithm for target tracking in wireless sensor networks

LU Xu<sup>1,2</sup>, CHENG Liang-lun<sup>1</sup>, LUO Shi-liang<sup>1</sup>

(1.College of Automatization, Guangdong University of Technology, Guangzhou 510006, China;

2.College of Automatization, Guangdong Polytechnic Normal University, Guangzhou 510665, China)

**Abstract:** An efficient node scheduling algorithm can significantly improve both the perceived ability and the energy efficiency for wireless sensor networks. Based on the model analysis of network node power consumption, an energy-efficient adaptive node scheduling algorithm for target tracking (ANSTT) is proposed for wireless sensor network. The algorithm automatically adjusts the working mode of sensor nodes according to the perceived ability on moving target and the relative remaining energy level (RREL) of nodes. Experiment results show that the proposed ANSTT algorithm effectively reduces system energy consumption and prolongs network lifetime while maintaining low detection delay and high target perceived rate.

**Key words:** wireless sensor networks; node scheduling; target tracking; adaptive; energy-efficient

### 1 引言

无线传感器网络能够协作地实时监测、采集和处理节点分布区域内的各种环境或监测对象的信息, 并将处理后的数据传送到网络中的特定位置<sup>[1]</sup>。由于无线传感器网络的分布式特性, 其对硬件和软件的设计都有特别的要求, 其中最主要的一点是降低能耗。降低能耗的最有效办法是利用最少的节点覆盖需求区域, 这就是节点调度算法需要完成的工作<sup>[2]</sup>。无线传感器网络通常具有节点密度高且随机分布的特点, 因此合理的节点调度算法设计可以有

效节约系统能量, 在目标跟踪型传感器网络中尤为如此。

当前, 研究者提出了一些基于无线传感器网络的目标跟踪方法<sup>[3~10]</sup>, 部分方法的研究是针对节点调度算法进行的。Wang 等<sup>[6]</sup>提出一种高能效目标跟踪节点选择 (EENSTT, energy-efficient node selection for target tracking) 算法, 该算法通过基于粒子滤波的单节点处理方法计算目标检测概率, 然后根据信息实体和节点剩余能量构建协作目标跟踪的目标函数。Vasanthi 等<sup>[7]</sup>提出一种目标跟踪节能调度 (EST, energy saving schedule for target tracking)

收稿日期: 2014-03-11; 修回日期: 2014-08-25

基金项目: 国家自然科学基金—广东联合基金资助项目(U1201251, U0935002); 省部产学研结合基金资助项目(2010B090400211)

**Foundation Items:** The Key Project of NSFC-Guangdong Union Foundation (U1201251, U0935002); The Combination Project of Guangdong Province and Ministry of Education (2010B090400211)

算法, 该算法中基站位于监测区域中心, 靠近基站的节点为了节约能量具有较长的休眠时间, 处于边界附近的节点一直处于活动模式, 当有目标进入监测区域时, 由处于边界的节点来唤醒其他休眠的邻居节点。该算法对于直接出现在监测区域内部的目标, 具有较差的感知能力。Cheng 等<sup>[8]</sup>将超声波传感器间干扰退避问题转化为信道共享问题, 提出一种用于目标跟踪型超声波传感器网络的基于分布式饱和度的调度算法 (DSDA, distributed saturation degree based algorithm), 算法中所有节点可在本地处理从邻居节点收集的信息。Jiang 等<sup>[9]</sup>基于运动学和概率论设计一种目标预测方法, 再根据该方法来选择合理的唤醒节点, 据此提出了一种基于概率预测和睡眠调度 (PPSS, probability based prediction and sleep scheduling) 的协议。Cheng 等<sup>[10]</sup>利用节点与邻居节点对感知目标的监测结果, 对节点状态进行调度, 提出一种能耗可行的节点调度算法。

上述算法在采用不同方式实现目标跟踪的同时, 充分利用邻居节点个数、距离等信息, 减少网络中的覆盖冗余, 延长了网络的生存时间。但上述算法都未对传感节点的能耗模型进行详细分析, 未提出应对网络中节点能量变化的自适应机制。

本文针对目标跟踪型网络应用, 提出一种仅依靠节点的本地信息, 高效地完成调度运行的无线传感器网络自适应目标跟踪节点调度 (ANSTT, adaptive node scheduling for target tracking) 算法。

## 2 问题描述

假设监测区域为一个半径为  $r_{\text{mon}}$  m 的平面圆形区域, 监测目标为单一点状目标, 以大小为  $v$  的速度在监测区域中匀速移动。监测区域中随机部署  $n$  个传感节点, 每个节点的有效感知半径为  $r_{\text{sen}}$  m, 只要监测目标和某一处于活动模式的节点的距离小于  $r_{\text{sen}}$  m, 目标就能被该节点成功感知。

本文中传感节点有 5 种工作模式: 配置模式、休眠模式、只听模式、预活动模式和活动模式。配置模式用于节点的初始化, 只在节点部署之前使用。休眠模式使节点处于待机状态, 节点的内部时钟振荡器将停止工作, 此模式下节点的能耗极低。只听模式是一种静音模式, 处于该模式下的节点将其感知模块关闭, 不主动发送任何报文, 仅监听网

络中的通信请求。活动模式是节点的标准工作模式, 在该模式下, 节点可以主动监测网络中的信息, 也可以主动发起信息的传输, 节点内部大部分的模块在该模式下都被启动, 因此该模式是能耗最大的工作模式。预活动模式是只听模式到活动模式的过渡模式, 但处于预活动模式的节点仍有一定几率转入休眠模式。

ANSTT 调度算法以“轮”为时间单位来运行, 每一轮的时间包括信息感知段和数据传输段。处于活动模式的传感节点在信息感知段中开启其 MCU 和感知模块, 在数据传输段中开启其 MCU 和通信模块。传感节点的基本结构如图 1 所示。

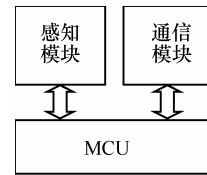


图 1 传感节点结构

传感节点的 MCU 实际也有多种操作模式, 每种操作模式的能耗水平都不同, 但本文中可以简单认为 MCU 有 2 种操作模式, 即正常模式 (W) 和待机模式 (S)。感知模块和通信模块中只要有一个在工作, MCU 就要处于正常模式。只有当传感节点进入休眠模式, MCU 才会进入待机模式, 以便节约能量。因此 MCU 在正常模式下的能耗为工作时间  $t_{\text{MCU\_WORK}}$  的正比例函数, 待机模式切换为正常模式所耗能量是一个固定值  $E_{\text{MCU\_S}\rightarrow\text{W}}$ , 因此 MCU 从待机到启动整个过程中的总能耗为

$$E_{\text{MCU}}(t_{\text{MCU\_WORK}}) = E_{\text{MCU\_S}\rightarrow\text{W}} + V_{\text{S}} I_{\text{MCU\_WORK}} t_{\text{MCU\_WORK}} \quad (1)$$

其中,  $I_{\text{MCU\_WORK}}$  表示 MCU 在正常模式下的电流值,  $V_{\text{S}}$  是 MCU 的供电电压。

假设在一个周期的时间  $T$  内, MCU 启动正常模式的次数不超过 1 次, 那么 MCU 在周期时间  $T$  内的能耗为

$$E_{\text{MCU}}(T, t_{\text{MCU\_WORK}}) = \begin{cases} E_{\text{MCU\_S}\rightarrow\text{W}} + V_{\text{S}}(I_{\text{MCU\_WORK}} - I_{\text{MCU\_STBY}})t_{\text{MCU\_WORK}} + V_{\text{S}}I_{\text{MCU\_STBY}}T, & t_{\text{MCU\_WORK}} > 0 \\ V_{\text{S}}I_{\text{MCU\_STBY}}T, & t_{\text{MCU\_WORK}} = 0 \end{cases} \quad (2)$$

其中,  $I_{\text{MCU\_STBY}}$  为 MCU 在待机模式下的电流值。

通信模块也有多种操作模式,包括待机模式、空闲模式、发送模式和接收模式,每种操作模式的能耗水平各有不同。本文中节点之间通信采用的MAC协议是基于 $\mu$ -MAC协议<sup>[11]</sup>改进的 $A\mu$ -MAC协议<sup>[12]</sup>。 $A\mu$ -MAC可以有效避免冲突、降低能耗,解决了 $\mu$ -MAC中动态拓扑适应性及时钟同步问题,并提供了良好的流量自适应性。 $A\mu$ -MAC的信道结构包含竞争期和无竞争期,竞争期采用分时槽的随机竞争接入方式,为保证分组99%传达,重发次数可达7次。在此基础上,通信模块发送报文时的电流变化曲线如图2所示。

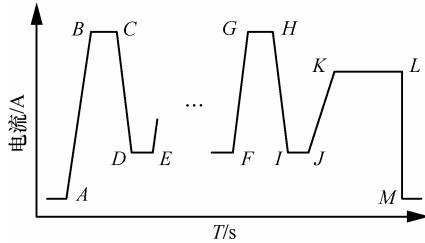


图2 通信模块发送报文时的电流变化曲线

$t_{TM\_A \rightarrow B}$ 是通信模块从待机模式转换为接收模式的延时时间,  $t_{TM\_B \rightarrow C}$ 、 $t_{TM\_G \rightarrow H}$ 是竞争期每次竞争持续的时长  $t_{TM\_COMP}$ ,  $t_{TM\_C \rightarrow D}$ 、 $t_{TM\_H \rightarrow I}$ 是通信模块从接收模式转换为空闲模式的延时时间,  $t_{TM\_D \rightarrow E}$ 、 $t_{TM\_I \rightarrow J}$ 是每次竞争后的空闲退避时间  $t_{TM\_IDLE}$ 。 $t_{TM\_J \rightarrow K}$ 是通信模块从空闲模式转换为发送模式的延时时间  $t_{TM\_IDLE \rightarrow TX}$ , 发送报文的时间  $t_{TM\_TX}$  为  $t_{TM\_K \rightarrow L}$ 。通信模块从开启到待机之间总的持续时间  $t_{TM\_TX\_TOT}$  为  $t_{TM\_A \rightarrow M}$ 。

$I_{TM\_D \rightarrow E}$ 、 $I_{TM\_I \rightarrow J}$ 是通信模块在空闲模式下的电流值  $I_{TM\_IDLE}$ 。 $I_{TM\_B \rightarrow C}$ 、 $I_{TM\_G \rightarrow H}$ 是通信模块在接收模式下的电流值  $I_{TM\_RX}$ ,  $I_{TM\_K \rightarrow L}$ 是通信模块在发送模式下的电流值  $I_{TM\_TX}$ 。通信模块的发射功率是可编程的,需根据通信距离的大小设置不同的发射功率,因此  $I_{TM\_TX}$  也不同。用  $E_{TM\_COMP}$  表示每次竞争花费的总能耗,  $n_{COMP}$  为竞争次数。因此,通信模块发送一个报文的总能耗可表示为

$$\begin{aligned} E_{TM\_TX}(t_{TM\_TX}, n) &= n_{COMP}(E_{TM\_COMP} + V_S I_{TM\_IDLE} t_{TM\_IDLE}) + \\ &\frac{1}{2} V_S (I_{TM\_TX} + I_{TM\_IDLE}) t_{TM\_IDLE \rightarrow TX} + V_S I_{TM\_TX} t_{TM\_TX} \end{aligned} \quad (3)$$

通信模块在监视通信信道到接收信息的电流变化曲线如图3所示。

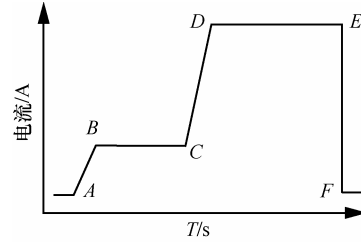


图3 通信模块接收信息时的电流变化曲线

$t_{TM\_A \rightarrow B}$ 是通信模块从待机模式转换为空闲模式的延时时间,  $t_{TM\_B \rightarrow C}$ 是通信模块处于空闲模式的时间  $t_{TM\_IDLE}$ ,  $t_{TM\_C \rightarrow D}$ 是通信模块处于接收模式的时间  $t_{TM\_RX}$ 。通信模块从待机模式转为空闲模式、从空闲模式转为接收模式的能耗都是固定值,分别用  $E_{TM\_RX\_S \rightarrow I}$  和  $E_{TM\_RX\_I \rightarrow R}$  表示。那么,通信模块在整个监视通信信道到接收信息过程中的能耗为

$$E_{TM\_RX}(t_{TM\_RX}) = E_{TM\_RX\_S \rightarrow I} + V_S I_{TM\_IDLE} t_{TM\_IDLE} + E_{TM\_RX\_I \rightarrow R} + V_S I_{TM\_RX} t_{TM\_RX} \quad (4)$$

在一个运行周期  $T$  内,传感节点的通信模块可能发生3种情况:①发送感知信息;②监视通信信道;③休眠。因此,在周期时间  $T$  内通信模块的能耗可表示为

$$E_{TM}(T, t_{TM+TX}, n, t_{TM\_RX}) = \begin{cases} n_{COMP}(E_{TM\_COMP} + V_S I_{TM\_IDLE} t_{TM\_IDLE}) + \\ \frac{1}{2} V_S (I_{TM\_TX} + I_{TM\_IDLE}) t_{TM\_IDLE \rightarrow TX} + \\ V_S I_{TM\_TX} t_{TM\_TX}, & t_{TM\_TX} > 0 \\ E_{TM\_RX\_S \rightarrow I} + V_S I_{TM\_IDLE} t_{TM\_IDLE} + \\ E_{TM\_RX\_I \rightarrow R} + V_S I_{TM\_RX} t_{TM\_RX}, & t_{TM\_RX} > 0 \\ V_S I_{TM\_STBY} T, & t_{TM\_TX} = t_{TM\_RX} = 0 \end{cases} \quad (5)$$

其中,  $I_{TM\_STBY}$ 是通信模块在待机模式下的电流值。

感知模块工作时的电流变化和MCU类似。感知模块从待机模式转为正常模式耗时  $t_{SEN\_S \rightarrow W}$ , 正常模式维持时间为  $t_{SEN\_WORK}$ 。感知模块在待机模式和正常模式下的工作电流分别为  $I_{SEN\_STBY}$  和  $I_{SEN\_WORK}$ 。感知模块从待机模式转为正常模式的能耗为一个固定值  $E_{SEN\_S \rightarrow W}$ 。假设在1轮当中,感知模块启动正常模式的次数不超过1次,那么其在周期时间  $T$  内的能耗为

$$E_{SEN}(T, t_{SEN\_WORK}) = \begin{cases} E_{SEN\_S \rightarrow W} + \\ V_S I_{SEN\_WORK} t_{SEN\_WORK}, & t_{SEN\_WORK} > 0 \\ V_S I_{SEN\_STBY} T, & t_{SEN\_WORK} = 0 \end{cases} \quad (6)$$

因此, 在一个运行周期  $T$  内, 传感节点的总能耗可表示为

$$\begin{aligned} E_{TOT}(T, t_{MCU\_WORK}, t_{TM\_TX}, n_{COMP}, t_{TM\_RX}, t_{SEN\_WORK}) \\ = E_{MCU}(T, t_{MCU\_WORK}) + E_{TM}(T, t_{TM\_TX}, n_{COMP}, t_{TM\_RX}) + \\ E_{SEN}(T, t_{SEN\_WORK}) \end{aligned} \quad (7)$$

本文的研究都是基于以上能耗模型进行。下面定义3个评价指标用来评估系统性能。

1) DD (detection delay): 当一个感知目标同时处于  $k$  个传感节点的感知范围内时, 称该感知目标属于  $k$ -coverage<sup>[13]</sup>。本文设定的网络应用中需要确定感知目标的位置, 根据三点定位原理, 为了确定目标位置, 至少需要目标处于 3-coverage 状态。本文定义的第一个评价指标是感知延时, 指感知目标从出现在监测区域起到其第一次处于 3-coverage 状态之间经历的时长。感知延时小的系统能够快速获取目标的位置信息, 有利于应用服务的开展和周边资源的分配。

2) TPR (target perceived rate): 本文定义的第2个评价指标为目标感知率, 指感知目标处于 3-coverage 状态的时间和感知目标处于整个监测区域内的总时间的比值。

3) TECA (total energy consumption in across): 本文定义的第3个评价指标是经过监测区域圆心的感知目标, 在经过整个监测区域过程中的网络的总能耗, TECA 可以反映系统的能效。

本文提出的无线传感器网络高效目标跟踪节点调度算法的主要目标就是在保持较低的 DD 和较高的 TPR 的前提下, 尽量降低 TECA。

### 3 ANSTT 算法描述

#### 3.1 ANSTT 算法的运行机制

在算法初始化阶段, 所有传感节点首先启动只听模式, 开始监听无线通信信道。通过监听信道, 节点可以收集邻居节点的信息, 包括2个参数。

1) 感知能力 (SA, sensor ability): 一个处于活动模式的传感节点  $i$  会在信息感知段  $t_{sen}$  内感知其周围环境, 如果在  $t_{sen}$  内捕捉到一个感知目标, 节点  $i$  就将其感知能力  $SA_i$  置1, 否则将  $SA_i$  置0。

2) 相对剩余能量水平 (RREL, relative remaining energy level): 设在当前轮中节点  $i$  的剩余能量为  $E_i$ , 网络中所有处于活动模式的节点中剩余能量最多的节点  $j$  的剩余能量为  $E_j$ , 则节点  $i$  的相

对剩余能量水平

$$RREL = E_i / E_j \quad (8)$$

$RREL$  的取值范围是  $[0, 1]$ ,  $RREL=0$  表示节点已死亡,  $RREL=1$  时表示该节点剩余能量是所有活动节点中剩余能量最多的。

节点在确定自己的 SA 和 RREL 之后, 将这些信息封装成一个固定格式的报文, 并在通信周期内把报文向周围广播。处于只听模式的传感节点利用收集到的邻居节点的信息来计算以下概率值:

$$p(n_{SA=0}) = \begin{cases} 0, n_{SA=0} = 0 \\ RREL \left(1 - \frac{1}{n_{SA=0}}\right)^3, 0 < n_{SA=0} \leq n_t \\ 0, n_{SA=0} > n_t \end{cases} \quad (9)$$

其中,  $n_{SA=0}$  是感知能力为0的邻居节点的个数,  $n_t$  是一个设定的邻居节点数阈值, 传感节点通过  $n_t$  值的调节来控制发送功率。在当前轮结束前, 处于只听状态的传感节点将以  $p(n_{SA=0})$  的概率转入预活动模式。对于  $0 < n_{SA=0} \leq n_t$  时  $p(n_{SA=0})$  的取值函数的选取, 将通过第4节的仿真实验进行分析。

在 ANSTT 算法中, 处于活动模式的无线传感节点会将其自身的感知能力信息向邻居节点广播。在正常情况下, 处于该广播范围内的节点数会随着网络密度增加而增加, 这意味着整体网络能耗的增加。为了将广播范围内的节点数维持在一个稳定水平, 应该将广播节点的通信半径  $r_{comm}$  调整到能够保持其邻居节点数量近似为上述的阈值  $n_t$ 。

调节节点的发射功率可以得到合适的  $n_t$  值。根据网络规模和节点性能预先设定节点的最小通信半径  $r_{min}$  和最大通信半径  $r_{max}$ , 并在  $r_{min}$  到  $r_{max}$  之间取  $g$  个级别,  $g$  值的最优取值通过第4节的仿真实验进行分析。

本文通过以下方法得到合适的通信半径  $r_{comm}$ : 首先初始化传感节点, 将其通信半径设置为最小值  $r_{min}$ , 然后向周边广播一个带有自身身份信息的数据分组, 该数据分组带有密度检测的控制信息;  $r_{min}$  范围内的邻居节点接收到密度检测分组后, 回发一个应答分组, 分组内带有发送节点和应答节点的身份信息; 发送节点根据收到的应答分组的数量, 估算其周边的邻居节点的个数, 如果邻居节点个数小于  $n_t$ , 就将发射功率提高一个级别, 从而增大  $r_{comm}$ ;

然后重复步骤上述过程，直至邻居节点数量大于或等于  $n_t$ ，此时得到的通信半径即为最终值。

$r_{\text{comm}}$  是在数据链路层进行设置的，上述方法用于定义阈值  $n_t$ ，然后将其送至数据链路层来使用。以上描述是初始化算法，在大部分应用中，存活节点的数量是随时间逐渐减少的，因此上面的算法在网络运行过程中每过固定时间便执行一次。每一次  $r_{\text{comm}}$  值被选定，节点将在其后的固定轮时间内一直采用该值来广播信息。该算法只是用于调整  $r_{\text{comm}}$  值，而和任何数据采集过程中涉及的通信半径无关。

对应上述转入预活动模式的概率，处于只听状态的传感节点转入休眠模式的概率是  $1-p(n_{SA=0})$ ，休眠时间  $t_s$  正比于该概率值

$$t_s = \lceil t_m(1 - p(n_{SA=0})) \rceil \quad (10)$$

$\lceil x \rceil$  指比  $x$  大的最小整数。从式(9)和式(10)可以看出，如果一个传感节点的大部分邻居节点的感知能力为 1 且相对剩余能量水平较高，那么该节点将会有较长的休眠时间，反之亦然。这就是说，当节点  $i$  的大部分邻居节点都不能成功感知目标，那么这个目标实际就处于该节点的感知范围，而不是其邻居节点的感知范围内，为了增加成功感知目标的概率，节点  $i$  应该缩短其休眠时间，增加进入活动模式的机会，但是这个增加的概率也受节点本身的剩余能量影响。相应地，如果节点  $i$  的大部分邻居节点的感知结果为 1，那么感知目标很可能已经被足够多的节点感知到，此时如果再将节点  $i$  转入活动模式，不但无法有效提高 TPR，从总体上来说还会增加额外的系统能耗。因此，此时节点  $i$  应该延长其休眠时间来节约能量。基本上只要符合上面所说的逻辑，式(9)也可以用其他的单向递减函数来替代，本文之所以采用上述形式是因为这个公式具有很好的性能。将在第 4 节中探讨采用不同形式的式(9)对本文的节点调度算法的性能的影响。

在每一轮的最后，感知能力为 0 的节点转入休眠模式，感知能力为 1 的节点继续维持活动模式。转入休眠模式的节点随机休眠一段时间  $t_s$ ， $t_s$  的取值范围是 0 到系统规定的最大休眠时间  $t_m$ ，每个节点内部有定时器来完成这个任务。休眠时间结束后，节点转入只听模式。ANSTT 算法的节点状态转换如图 4 所示。

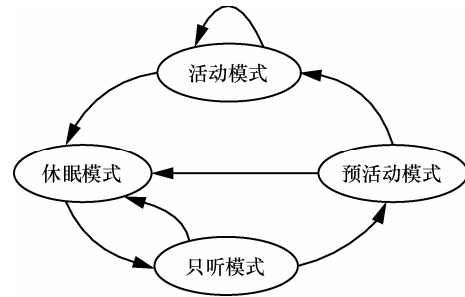


图4 ANSTT 算法的节点状态转换

### 3.2 ANSTT 算法分析

在 ANSTT 算法中，节点在第  $t$  轮的状态分布只和第  $t-1$  轮的节点状态有关，而和第  $t-1$  轮之前的第  $1, 2, \dots, t-2$  轮都无关，这样一个离散随机分布过程，是典型的马尔可夫过程<sup>[14]</sup>，这个离散随机序列可用马尔可夫链来描述<sup>[15]</sup>。

本文中每个传感节点的运行状态都可以由一个马尔可夫链来描述，可表示一系列的状态  $X_t$ ，即  $X_1, X_2, \dots$ 。这些状态的值都包含在集合  $U$ （状态空间）中，这个集合由 4 个主要的子集  $S$ 、 $L$ 、 $P$  和  $W$  组成，分别表示节点处于休眠模式、只听模式、预活动模式和活动模式。

子集  $S$  由  $m$  个状态值组成，状态值分别表示为  $s_1, s_2, \dots, s_m$ ，每一个状态表示传感节点在休眠模式时不同的剩余休眠时间。子集  $L$  由  $n$  个状态值组成，分别为  $l_1, l_2, \dots, l_n$ ，每个状态表示传感节点在只听模式监测到的感知结果为 1 的邻居节点个数。子集  $P$  只有 1 个状态值，即  $p_1$ 。子集  $W$  由小子集  $W_s$  和状态值  $W_u$  组成。 $W_s$  表示在第  $t$  轮成功感知到目标的处于活动模式的传感节点，能够保持活动模式直至感知失败的轮数。这意味着活动节点将持续处于活动模式 1 到  $2r_{\text{sen}}/v$  轮，这取决于他们的位置和目标的移动方向之间的关系。因此， $W_s$  将由  $e=2r_{\text{sen}}/v$  个状态值组成，分别是  $w_{s1}, w_{s2}, \dots, w_{se}$ 。 $W_u$  表示在第  $t$  轮不能成功感知到目标的处于活动模式的传感节点。因此，节点的运行状态马尔可夫链如图 5 所示。

因此，一个最大休眠时间为  $t_m$  轮、邻居节点数为  $n_t$ 、活动模式持续时间为 0 到  $e$  轮的传感节点，将有一个维度为  $|D|=t_m+n_t+e+3$  的集合。对于处于休眠模式的传感节点，在子集  $S$  内部转换的概率为

$$P(s_{t-x} | s_{t-1-y}) = \begin{cases} 1, & x = y - 1 \\ 0, & \text{其他} \end{cases} \quad (11)$$

其中， $1 \leq y \leq t_m$ 。

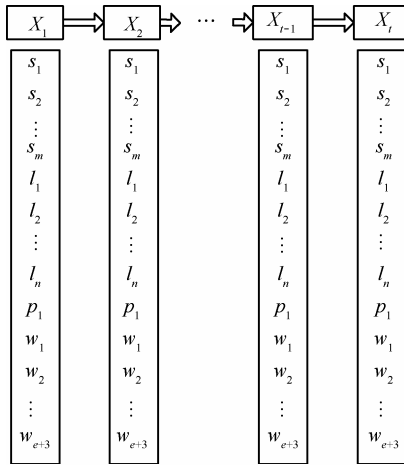


图5 节点运行状态的马尔可夫链示意

利用分布向量来评估 ANSTT 算法的性能。分布向量  $\omega_t$  是一个  $|D|$  维向量, 其每一个元素表示一个传感节点转入某一特定模式的概率, 所有项的总和为 1。根据 ANSTT 算法的运行机制, 对于  $A \in D$ ,  $\omega_t$  的项可表示为

$$P(A) = \begin{cases} \frac{1}{t_m}, & A = s_{1-x}, x = 1, 2, \dots, n_t \\ 0, & \text{其他} \end{cases} \quad (12)$$

ANSTT 算法的分布向量可由以下迭代方程获得

$$\omega_t = Q_{t-1} \omega_{t-1} \quad t = 1, 2, \dots \quad (13)$$

每次迭代过程中, 从  $\omega_t$  中获得  $P(w_{s_t})$ 、 $P(w_{u_t})$  和  $P(l_t)$ , 另外  $P(w_{s_t})$  还用来计算  $Q_t$ 。

当处于休眠模式的节点休眠时间结束, 在下一轮将转入只听模式。处于  $s_{t-1}$  状态值的节点转入只听模式的概率值取决于以下概率值

$$P(w_{t-1-j}) = \sum_{i=1}^c P(w_{t-1-j-i}) \quad (14)$$

其中,  $P(w_{t-1-j-i})$  可由分布向量  $\omega_{t-1}$  求得。那么转换概率可表示为

$$P(l_{t-x} | s_{t-1-x}) = C_b^x (1 - P(w_{t-1-j}))^{b-x} (P(w_{t-1-j}))^x \quad (15)$$

其中,  $x = 1, 2, \dots, n_t$ 。

处于只听模式的节点可能进入休眠模式或预活动模式, 进入休眠模式的概率为

$$P(s_{t-y} | l_{t-1-x}) = \begin{cases} 1 - v(x), & y = (1 - v(x))t_m \\ 0, & \text{其他} \end{cases} \quad (16)$$

其中,  $x = 1, 2, \dots, n_t$ 。

转入预活动模式的概率为

$$P(p_1 | l_{t-1-x}) = v(x) \frac{\pi R^2}{\omega} \quad (17)$$

处于预活动模式的节点转入休眠模式的概率为

$$P(s_{t-x} | p_1) = \sum_{i=1}^x P(l_{t-1-i}) P(l_{t-1-x}) \quad (18)$$

处于预活动模式的节点转入活动模式但未成功感知到目标的概率为

$$P(w_{u-t-x} | p_1) = P(p_1) \left( 1 - \frac{\pi R^2}{\omega} \right) \quad (19)$$

在前面讲过, 成功感知的节点将保持活动模式一段时间, 时间长短取决于和感知目标之间的位置和方向相关性。对于一个速度为  $v$  m/round 的目标, 有效感知区域也要以同样的速度移动。因此, 转换概率  $P(w_{t-x} | p_1)$  取决于可感知区域在不同轮当中的重叠面积, 用  $f(\Delta t)$  来表示时间差为  $\Delta t$  时的重叠面积, 即

$$f(\Delta t) = 2 \left( \arccos \left( \frac{v\Delta t}{2R} \right) \right) R^2 - \left( \sqrt{R^2 - \left( \frac{v\Delta t}{2} \right)^2} \right) v\Delta t \quad (20)$$

处于预活动模式的节点转入活动模式并成功感知目标的概率为

$$P(w_{s-t-x} | p_1) = (1 - \frac{\pi R^2}{\omega}) (f(q-1) - f(q)) \quad (21)$$

其中,  $q = 1, 2, \dots, e$ 。

最后, 子集  $W_{u,t-1}$  内的节点将以相等的概率转入  $S_{t,x}$ , 即

$$P(s_{t-x} | w_{u,t-1}) = \frac{1}{t_m} \quad (22)$$

其中,  $x = [1, t_m]$ 。

利用所有上述转换概率, 构建一个转换矩阵  $Q_{t-1}$ ,  $Q_{t-1}$  是一个  $|D| \times |D|$  的方形矩阵, 其中每一个元素都是一个节点从第  $t-1$  轮的某一特定模式转入第  $t$  轮的另一种模式的转换概率。

在节点数为  $n$  的无线传感器网络中, 一个目标在第  $t$  轮中被至少  $m$  个节点同时感知的概率为

$$P(n, m, t) = 1 - \sum_{\tau=0}^m (C_{n-\tau}^n (1 - P(w_{s_{-t}}))^{n-\tau} (P(w_{s_{-t}}))^{\tau}) \quad (23)$$

根据式(23), 在 ANSTT 算法中, 节点数为  $n$  的无线传感器网络的 DD 可表示为

$$DD = \sum_{t=1}^{t_d} \left\{ t \left( \prod_{\sigma=0}^{t-1} P(n, 3, \sigma) P(n, 3, t) \cdot (C_{n-\tau}^n (1 - P(w_{s_{-t-1}}))^{n-\tau} (P(w_{s_{-t-1}}))^{\tau}) \right) \right\} \quad (24)$$

其中,  $t_d$  为目标横跨感知区域所需要的轮数。

同样, TPR 可表示为

$$TPR = \frac{\sum_{t=1}^{t_d} P(n, 3, t)}{t_d} \quad (25)$$

TECA 可由节点总数  $n$ 、分布向量和每种模式下的能耗总量来决定。在 ANSTT 算法中, 处于活动模式的节点每一轮消耗能量为

$$E_W = E_{TOT}(T, t_{TM\_TX\_TOT}, t_{TM\_TX}, n_{COMP}, 0, 0) + E_{TOT}(T, t_{SEN\_TOT}, 0, 0, 0, t_{SEN\_A}) \quad (26)$$

有  $h$  个活动邻居节点的处于只听模式的节点每一轮消耗

$$E_{L_h} = hE_{TOT}(T, t_{TM\_RX\_TOT}, 0, 0, t_{TM\_RX}, 0) \quad (27)$$

处于休眠模式的节点每一轮消耗

$$E_S = E_{TOT}(T, 0, 0, 0, 0, 0) \quad (28)$$

这样 TECA 可表示为

$$TECA = \frac{2r_{mon}n}{vTSR} \cdot \left\{ \frac{\sum_{t=1}^{t_d} E_W P(w_t) + \sum_{h=0}^b E_{L_h} P(l_{t_h}) + \sum_{i=0}^a E_S P(s_{t_i})}{t_d} \right\} \quad (29)$$

## 4 仿真和评价

本文所设计算法主要关注能够唤醒移动目标周围的足够数量的活动节点, 因此数据收集过程如成簇<sup>[16]</sup>、路由<sup>[17]</sup>和数据收集<sup>[18]</sup>等都不在本文考虑的范围。使用 Matlab 进行仿真实验, 监测场景为一个半径

为 200 m 的圆形区域, 基站位于圆心位置。感知目标的移动速度为 5 m/round。为了消除边界的影响, 目标只出现在  $r=160$  m 的感知区域内, 因此  $t_d=160 \times 2/5=64$ 。每个节点的初始能量为 10 000 J, 供电电压  $V_S$  为 3.3 V, 节点的有效传感半径  $r_{sen}$  为 40 m, 通信速率为 120 kbit/s。能耗模型中的参数如表 1 所示。

表 1 能耗模型参数

参数	设定值
$t_{TM\_S \rightarrow I}$	800 $\mu$ s
$t_{TM\_IDLE \rightarrow RX}$	150 $\mu$ s
$t_{TM\_RX \rightarrow IDLE}$	0 $\mu$ s
$t_{TM\_IDLE \rightarrow TX}$	150 $\mu$ s
$t_{TM\_TX \rightarrow IDLE}$	0 $\mu$ s
$t_{TM\_IDLE}$	3.2 ms
$I_{SEN\_WORK}$	12.5 s
$t_{MCU\_S \rightarrow W}$	3.4 ms
$t_{MCU\_W \rightarrow S}$	0 $\mu$ s
$t_{TM\_RX}$	14.7 ms
$t_{TM\_TX}$	1.2 ms
$t_{SEN\_S \rightarrow W}$	180 $\mu$ s
$I_{MCU\_WORK}$	7.8 mA
$I_{MCU\_STBY}$	0.005 mA
$I_{SEN\_WORK}$	4.7 mA
$I_{SEN\_STBY}$	0.002 mA
$I_{TM\_TX}$	7.6~17.6 mA
$I_{TM\_RX}$	17.6 mA
$I_{TM\_IDLE}$	0.38 mA
$I_{TM\_STBY}$	0.003 mA

### 4.1 评价函数 $p(n_{SA=0})$

在第 3 节中讲过, 可以用具有相似特性的其他函数来替换评价函数  $p(n_{SA=0})$ 。首先采用不同的评价函数进行仿真实验来评估 ANSTT 算法的 TECA。网络规模取  $n=600$ , 参数  $t_m$ 、 $n_t$  和  $g$  根据 2 组参数来调节。1)  $DD \leq 2$  个时槽、 $TPR \geq 97\%$ ; 2)  $DD \leq 4$  个时槽、 $TPR \geq 95\%$ 。实验采用 5 个评价函数, 如表 2 所示, 当  $n_{SA=0} > n_t$  时, 评价函数均为  $p(n_{SA=0}) = 0$ , 因此表 2 只列出  $n_{SA=0} \leq n_t$  时的函数。几个评价函数选择依据是能够起到算法所需约束作用的常用函数, 分别是指数函数、线性函数、平方函数、立方函数和四次函数。仿真结果如图 6 和图 7 所示。

表2 几种不同的评价函数  $p(n_{SA=0})$

序号	$n_{SA=0} \leq n_t$
1	$p(n_{SA=0}) = RREL \times 0.68^{n_{SA=0}}$
2	$p(n_{SA=0}) = RREL(1 - \frac{1}{n_{SA=0}})$
3	$p(n_{SA=0}) = RREL(1 - \frac{1}{n_{SA=0}})^2$
4	$p(n_{SA=0}) = RREL(1 - \frac{1}{n_{SA=0}})^3$
5	$p(n_{SA=0}) = (1 - \frac{1}{n_{SA=0}})^4$

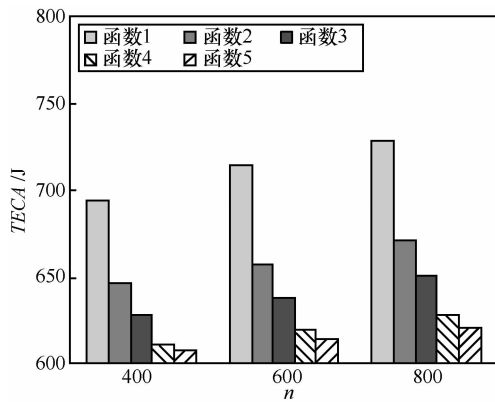


图6  $DD \leq 4$ 、 $TPR \geq 95\%$ 时不同评价函数的  $TECA$  值

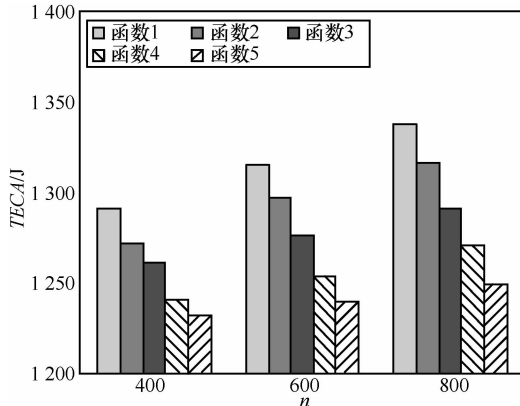


图7  $DD \leq 2$ 、 $TPR \geq 97\%$ 时不同评价函数的  $TECA$  值

由图6和图7可以看出,不同评价函数的  $TECA$  值有所不同,采用函数1的  $TECA$  值最大,函数2和函数3次之,函数4是 ANSTT 算法采用的评价函数,其  $TECA$  值较小。但从图中可看出,函数5比函数4的  $TECA$  值还要略小,本文之所以采用函数4而不采用函数5,是因为在评价函数中引入相对剩余能量水平  $RREL$  可有效平衡整个网络中节点的能耗,从而延长网络的整体寿命。另外,从图6和图7还可看出,对  $DD$  和  $TPR$  要求不同  $TECA$  的值也不同,  $DD$  越小、 $TPR$  越大,  $TECA$  就越高。

### 4.2 通信模块发送功率级别参数 $g$

通过仿真实验来研究节点的通信模块发送功率级别参数  $g$  对 ANSTT 算法性能的影响。把网络节点总数设为 600, 仿真 20 次来计算  $DD$ 、 $TPR$  和  $TECA$  的平均值, 每一次仿真都维持 500 轮。把参数  $n_t$  固定为 3, 通过不同的  $t_m$  来评价不同  $g$  值的网络性能。仿真结果如图8~图10所示。

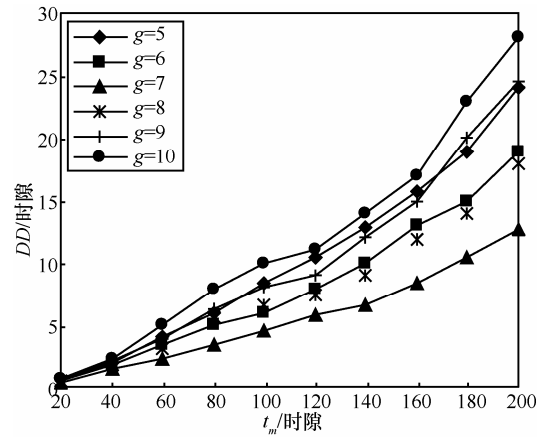


图8 不同  $g$  值时 ANSTT 算法的  $DD$

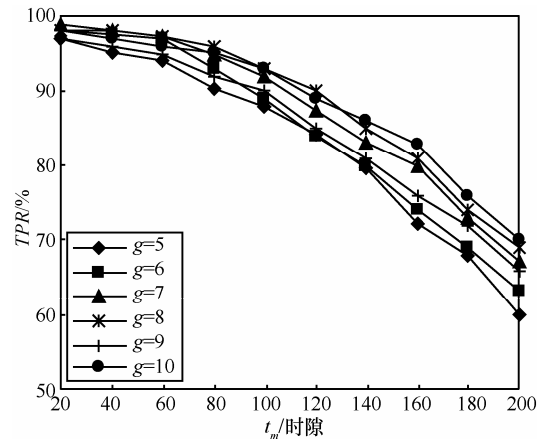


图9 不同  $g$  值时 ANSTT 算法的  $TPR$

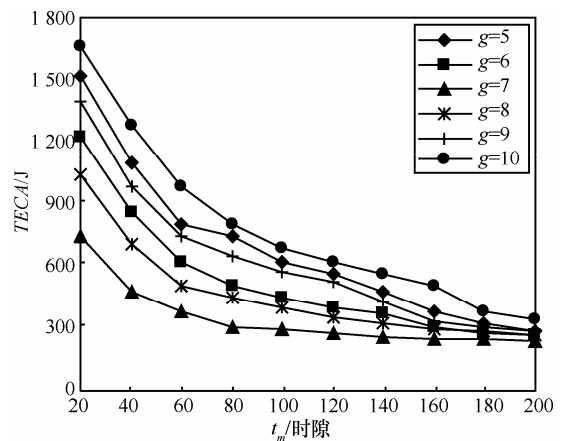


图10 不同  $g$  值时 ANSTT 算法的  $TECA$

图8是感知延时  $DD$  随节点最大休眠时间  $t_m$  变化的情况。由图8可以看出,系统的  $DD$  是随着  $t_m$  的增加而增加的,这是因为  $t_m$  增加,节点平均处于休眠模式的时间就增加,使系统的感知能力下降,影响了感知的速度。同时,当  $g$  值取7,也就是通信模块发送功率级别参数设为7时,系统的感知延时  $DD$  值比  $g$  值取5、6、8、9或10时都要小,可见  $DD$  值随  $g$  值的增加成U型曲线。这是由于随着  $g$  值的增加,对感知目标的监测反应应该越快,但  $g$  值越高,算法时间复杂度越高,导致算法延时增加,  $g$  值取7时,感知延时  $DD$  出现拐点。

图9是目标感知率  $TPR$  随  $t_m$  变化的情况。由图9可以看出,随着  $t_m$  的增大,网络的目标感知率是呈降低趋势的,这是因为  $t_m$  的增加,使节点处于休眠模式的时间增长,对感知目标的有效感知次数就减少了。从图9还可看出,随着  $g$  值的增大,  $TPR$  会相应地提高,这是由于  $g$  值越大,分级越详细,对目标的感知越精确,从而降低了目标丢失的概率。但也可以看出,  $g$  值取7以上时,  $TPR$  随  $g$  值增大而提高的程度已经不大。

图10是感知目标完全经过(经过圆心)整个监测区域时网络的总能耗  $TECA$  随  $t_m$  变化的情况。由图10可以看出,随着  $t_m$  的增大,  $TECA$  是不断降低的,这是因为  $t_m$  的增大增加了网络中传感节点处于休眠模式的时间,从而为节点节约了更多能量。同时,当  $g$  值取7,也就是通信模块发送功率级别参数设为7时,系统的  $TECA$  值比  $g$  值取5、6、8、9或10时都要小。随着  $g$  值的增加,系统对感知目标监测的精度增加,可以更加精确地控制节点感知目标时的能耗,那么系统的总能耗也本应不断降低。但随着  $g$  值的增大,同时增加了网络对感知目标的有效感知率,算法复杂度也有一定程度增加,综合作用下,使系统的总能耗无法有效降低。因此,仿真结果显示,  $TECA$  值随  $g$  值的变化成U型曲线,  $g$  值取7时,  $TECA$  值出现拐点。

综上所述,在上述仿真实验所采用的参数下,通信模块发送功率级别参数  $g$  取7时,网络的整体性能最优。

#### 4.3 感知延时和目标感知率

对 EENSTT 算法、EST 算法和 ANSTT 算法在相同网络环境下进行仿真,比较各算法的感知延时和目标感知率情况。网络节点总数设为600,仿真

20次来计算各算法  $DD$ 、 $TPR$  的平均值,每一次仿真都维持500轮。参数  $n_i$  固定为3,通过不同的  $t_m$  来测试各算法。EENSTT 算法和 EST 算法中的参数根据文献[6,7]中的最优参数进行设置。仿真结果如图11和图12所示。

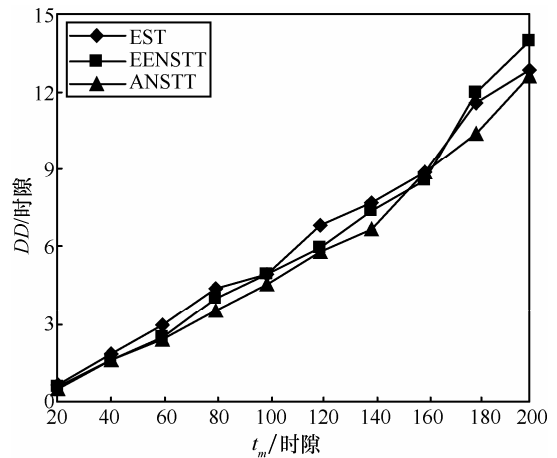


图11 各算法的感知延时  $DD$

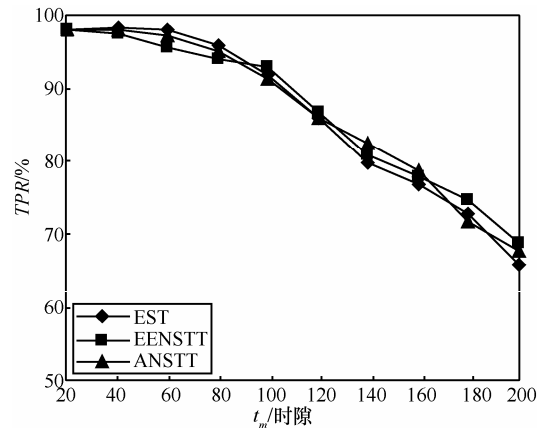


图12 各算法的目标感知率  $TPR$

由图11和图12可以看出,ANSTT 算法维持了 EST 和 EENSTT 接近的感知延时和目标感知率,在感知延时方面,ANSTT 算法相对于另外2种算法还略有优势。

#### 4.4 网络寿命

对 EENSTT 算法、EST 算法和 ANSTT 算法在相同网络环境下进行仿真,比较各算法的能耗情况。ANSTT 算法的  $n_i$  值设为3,  $t_m$  值设为40,  $g$  值设为7。EENSTT 算法和 EST 算法中的参数根据文献[6,7]中的最优参数进行设置。本文在不同规模的网络( $n=400$ 、 $600$ 、 $800$ )中,各仿真20次来计算各算法的存活节点数的平均值。仿真结果如图13~图15所示。

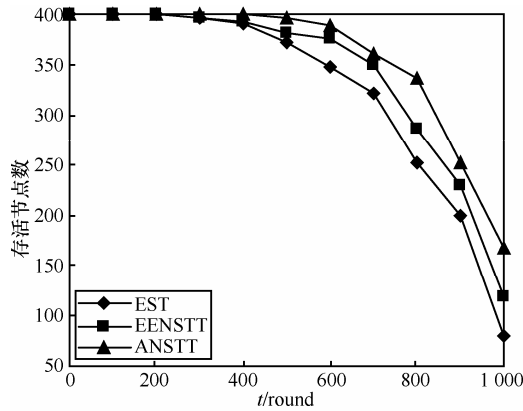


图 13  $n=400$  时各算法的存活节点数

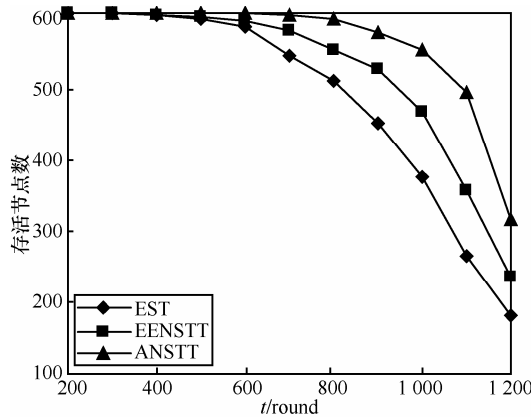


图 14  $n=600$  时各算法的存活节点数

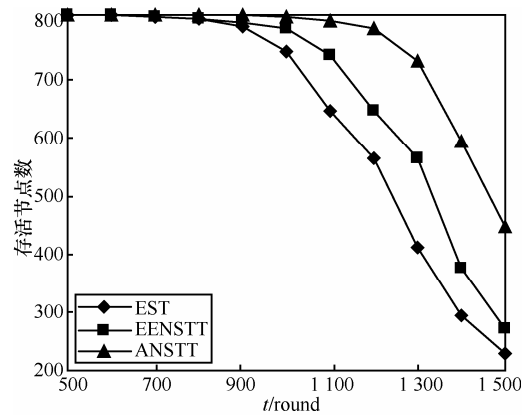


图 15  $n=800$  时各算法的存活节点数

由图 13~图 15 可以看出，随着网络的运行，ANSTT 算法中节点的死亡速度比 EENSTT 和 EST 算法慢，存活节点的数量比 EENSTT 和 EST 算法多，说明 ANSTT 算法在延长网络寿命方面相对于 EENSTT 和 EST 算法具有的优越性。这是由于 ANSTT 算法采用了自适应的应对节点能量变化的调节机制，使能量较低的节点较多地处于休眠模式，而能量较高的节点会有相对较大的概率处于活动模式，这就有效平衡了传感节点的能

耗。另外，从图 13~图 15 还可看出，网络节点总数越多，ANSTT 算法在网络寿命方面的优越性越明显，说明 ANSTT 算法更适合于大规模的网络应用，其自适应机制在节点密度较大的环境中所起的作用更大。

### 5 结束语

本文针对移动目标跟踪型的无线传感器网络应用，提出一种自适应的节点调度算法 ANSTT，并提出 3 个性能指标对 ANSTT 进行性能评价，分析了 3 个性能指标在 ANSTT 中的概率模型。仿真结果表明，相比于 EENSTT 和 EST 算法，ANSTT 算法在维持较低的感知延时、较高的感知效率的同时，可有效节约系统能量，平衡节点之间的能耗，延长网络寿命，尤其在节点密度较大的网络环境中，其性能优势更加显著。

### 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.  
SUN L M, LI J Z, CHEN Y, *et al.* Wireless Sensor Networks[M]. Beijing: Tsinghua University Press, 2005.
- [2] WANG L, XIAO Y. A survey of energy-efficient scheduling mechanisms in sensor networks[J]. Mobile Networks and Application, 2006, 11(5): 723-740.
- [3] CARLOS T, CARLOS L, AMOLDO D, *et al.* An integral model for target tracking based on the use of a WSN [J]. Sensors, 2013, 13(6): 7250-7278.
- [4] XU E Y, DING Z, DASGUPTA S. Target tracking and mobile sensor navigation in wireless sensor networks [J]. IEEE Transactions on Mobile Computing, 2013, 12(1): 177-186.
- [5] CHENG Y Q, WANG X Z, MORAN B, *et al.* Information geometry of target tracking sensor networks [J]. Information Fusion, 2013,14(3): 311-326.
- [6] WANG Y, WANG D H. Energy-efficient node selection for target tracking in wireless sensor networks[J]. International Journal of Distributed Sensor Networks, 2013, (2013): 1-6.
- [7] VASANTHI N A, ANNADURAI S. Energy saving schedule for target tracking sensor networks to maximize the network lifetime[A]. Proceedings of the 1st International Conference on Communication System Software and Middleware[C]. Delhi, India, 2006. 1-8.
- [8] CHENG P, ZHANG F, CHEN J M, *et al.* A distributed TDMA scheduling algorithm for target tracking in ultrasonic sensor networks[J]. IEEE Transactions on Industrial Electronics, 2013, 60(9): 3836-3845.
- [9] JIANG B, RAVINDRAN B, HYEONJOONG C. Probability-based prediction and sleep scheduling for energy-efficient target tracking in sensor networks[J]. IEEE Transactions on Mobile Computing, 2013,

- 12(4): 735-747.
- [10] CHENG C T, TSE C K, LAU F C M. An energy-aware scheduling scheme for wireless sensor networks[J]. IEEE Transactions on Vehicular Technology, 2010, 59(7): 3427-3444.
- [11] BARROSO A, ROEDIG U, SREENAN C.  $\mu$ -MAC: an energy-efficient medium access control for wireless sensor networks[A]. Proceedings of the 2nd European Workshop on Wireless Sensor Networks[C]. Washington D C, USA, 2005. 70-80.
- [12] 古连华, 程良伦等. A $\mu$ -MAC:一种自适应的无线传感器网络 MAC 协议[J].自动化学报, 2010, 36(1): 54-59  
GU L H, CHENG L L, *et al.* A $\mu$ -MAC: an adaptive MAC protocol for wireless sensor networks[J]. Acta Automatica Sinica, 2010, 36(1): 54-59.
- [13] GAO S, VU C T, LI Y S. Sensor scheduling for  $k$ -coverage in wireless sensor networks[J]. Mobile Ad-Hoc and Sensor Networks, 2006, 4325:268-280.
- [14] CHIASSERINI C F, GARETTO M. An analytical model for wireless sensor networks with sleeping nodes[J]. IEEE Transactions on Mobile Computing, 2006, 5(12): 1706-1718.
- [15] 刘克.实用马尔可夫决策过程[M]. 北京:清华大学出版社,2004.  
LIU K. Practical Markov Decision Process[M]. Beijing: Tsinghua University Press, 2004.
- [16] MOHAMMAD B, AHMAD A K, ALAA E A, *et al.* An energy-efficient threshold-based clustering protocol for wireless sensor networks[J]. Wireless Personal Communications, 2013, 70(1): 99-112.
- [17] SALEEM M, ULLAH I, FAROOQ M. BeeSensor: an energy-efficient and scalable routing protocol for wireless sensor networks[J]. Information Sciences, 2012, 200: 38-56.
- [18] FENG J, LU Y H, JUNG B, *et al.* Energy-efficient data dissemination using beamforming in wireless sensor networks[J]. ACM Transactions on Sensor Networks, 2013, 9(3): 31.

## 作者简介:



卢旭(1986-), 男, 江西修水人, 广东技术师范学院讲师, 广东工业大学博士生, 主要研究方向为物联网、无线传感器网络等。



程良伦(1964-), 男, 湖北黄石人, 广东工业大学教授、博士生导师, 主要研究方向为物联网与信息物理融合系统、RFID与传感器网络等。



罗世亮(1978-), 男, 江西信丰人, 广东工业大学博士生, 主要研究方向为物联网、信息物理融合系统。