A Light Weight Fault-Tolerant Event Detection Method in Wireless Sensor Networks *

DONG Chuanming[1], *LIU Kezhong*[1,2*], *LUO Guang*[2], *JIN Huting*[3]

(1. *School of Navigation, Wuhan University of Technology, Wuhan* 430063, *China*; 2. *School of Information Engineering, Wuhan University of Technology, Wuhan* 430070, *China*; 3. *Zhejiang Institute of Communications, Hangzhou* 311112, *China*)

Abstract: Due to harsh environment and limited resources, nodes are prone to missed alarm and false alarm errors when in the process of running, so fault-tolerant event detection methods attract much attention in recent years. But traditional algorithms showed high computational complexity and high energy consumption. Aim at this, we adopt the C_{th}/m detection idea in radar target detection, and propose a detection method based on spatial-temporal characteristics. By observing the agreement between sample's changing rate and the spatial-temporal characteristics, nodes can determine whether event happens or error occurs. During the detecting process, nodes are only required to do m times distance comparison and one time neighbor query, then it can make decision, reflecting the light weighting and low power consumption characteristics of the method. Simulation results show that the method also has low missed alarm rate and false alarm rate, and high fault recognition rate. In addition, as credibility adaptive adjustment mechanism is introduced, the method can ensure event detection probability maintained at a high level in the iterative process and not deteriorate over time.

Key words: wireless sensor networks; event detection; fault tolerant; spatial-temporal characteristics

EEACC: 6150P **doi**: 10.3969/j.issn.1004-1699.2014.01.025

无线传感器网络环境下的一种轻量级事件容错检测算法 *

董传明[1], 刘克中[1,2*], 罗 广[2], 金湖庭[3]

(1. 武汉理工大学航运学院, 武汉 430063; 2. 武汉理工大学信息工程学院, 武汉 430070; 浙江交通职业技术学院, 杭州 311112)

摘 要: 由于节点经常布置于恶劣的环境中以及节点自身资源的有限性, 节点在运行过程中容易发生漏警错误和虚警错误, 因此容错的事件检测方法受到了广泛的重视与研究。但传统方法在性能上表现出计算复杂度高和能耗大的特点, 针对此, 引入雷达目标检测中 C_{th}/m 的检测思想, 提出一种基于时空特性的检测方法。该方法通过观察节点采样值数据变化率与时间特性相似度来判断是事件发生还是节点出错。方法在一次迭代过程中, 节点只需做 m 次距离比较和一次邻居状态查询, 根据节点间的空间特性做出决策, 表现出轻量级和低能耗的特点。仿真表明, 方法还具有较低的漏警率和虚警率以及较高的故障识别率。此外, 由于引入了节点可信度自适应调整机制, 方法能够保证事件检测概率在迭代过程中一直保持较高水平, 而不随时间恶化。

关键词: 无线传感器网络; 事件检测; 容错; 时空特征

中图分类号: TP391.9 **文献标识码**: A **文章编号**: 1004-1699(2014)01-0135-07

无线传感器网络因价格低廉、配置灵活、抗毁性强等一系列特点[1]被广泛应用于众多事件检测, 如火灾检测、化学物质泄漏检测等。但由于网络通常部署无人看守的地方, 节点容易受恶劣环境的干扰, 加之节点自身资源的有限性, 往往会引起节点信息采集错误。传感器节点容易产生的错误主要有漏警错误和虚警错误。漏警错误指事件发生了但节点未检测到; 虚警错误指事件未发生但节点虚报有事件发生。在实际应用中, 我们期望这两类错误越小越好。

在无线传感器网络用于事件检测时, 确保采集数据的可靠性和消除检测错误的干扰, 是事件检测需要解决的一个关键问题。张树奎等[2]提出了一种新的分布式容错检测算法, 通过构建的融合树获得事件检测的属性值, 将检测错误率控制在可接受的范围内。Nguyen-Thanh N 等[3]基于证据理论设计了一种自适应的事件探测器, 该探测器通过比较采样数据与经验分布对比, 根据其符合程度来验证事件发生的假设成立与否, 从而实现事件检测。文献[4]针对感兴趣的

项目来源: 国家自然科学基金项目(51279151); 中央高校基本科研业务费专项资金项目(2012-IV-28, 2013-ZY-104); 浙江省交通运输厅科技计划项目(2012w05, 2012w06)

收稿日期: 2013-10-08 **修改日期**: 2013-12-18

事件周围失效节点的识别问题,提出局部失效节点识别和可容错的事件边缘监测算法。这些方法能够获得较好的检测效果,对事件区域检测具有一定的借鉴意义,但这些方法未考虑数据在容错检测过程中的计算复杂度,以及如何在保证纠错效果的同时延长网络的生存周期等问题。

一般情况下,检测环境在短时间内不会发生改变,传感器节点在较短时间内的检测值也是彼此相关的。因而可以利用节点的时间相关性对节点出现的错误信息进行容错处理。Ould-Ahmed-Vall E 等[5]考虑了一种容错的事件检测方法,节点通过统计历史数据来确定自己的出错概率,从而决定自己在邻域投票中所占的权重。但是,利用时间容错检测也有一定的局限性,它只能对暂时性错误进行纠正,而对于永久性错误的节点产生的错误束手无策;同时,由于节点是离散分布在事件检测区域,而时间相关性是针对每一节点的容错处理,单个节点有时不能对事件信息进行完整表达,因而时间相关性的容错只能算是数据的预处理,不能达到事件容错的目的。

相邻节点在检测环境中是具有空间相关性,利用相邻节点间的相关的冗余信息对节点事件检测进行容错处理。文献[6-7]都利用邻居节点协作,通过共同决策形成具有故障容忍能力的故障检测方法。黄日茂等[8]采用一种基于邻居节点数据分析的 WSN 故障检测方法,将节点数据与覆盖范围内可信邻居节点的数据进行比较分析,判定节点是否故障。文献[9]提出了一种空间相关的事件容错检测方法,当某节点检测到数据偏离了正常值,或者邻居节点的检测数据超过了预定的阈值,则可能是某事件发生或是节点出现故障。以上算法要求邻居节点间相互通信,会消耗较多的能量,尤其在算法较为复杂时,网络寿命会受到严重影响。

随着时间的变化,当节点采集到的数据保持平稳,一般认为该区域对传感器事件检测来说是时空相关的[10],本文提出一种分布式的轻量级的事件容错检测方法,该方法通过匹配节点采样值变化率与时间特性的相似度来实现事件检测。方法实现过程中,节点只需要进行 m 次距离比较并询问一次邻居节点就可以实现检测,因此算法是轻量级的。在询问邻居过程中,由于节点是 m 个采样间隔才交换一次数据且每次交换的数据是节点决策后的二元信息,所以算法是高能效的。

1 事件检测的过程

事件是指每个事件都可以看作由若干个事件属性组合而成的,如火灾这个事件由温度、烟雾浓度、光照强度等属性组成。事件属性是事件本身固有的性质,是事件最基本的、必然的、不可再分的特性,它可以从某个方面表现事件。火灾事件中,温度、烟雾浓度、光照强度是不可再分的火灾事件本身的特性,温度升高,烟雾浓度升高,光照强度的变化都是火灾的表现。本文对事件的检测是一个从模糊到清晰过程,其分为 3 个步骤:发现准事件,基本认定事件,核实精确事件。

因此,事件检测程序包括 3 个环节,第 1 个环节为发现准事件,节点自身检测环节,节点通过匹配采样值变化率与模式的关系初步判断事件是否发生。为了进一步确认事件,节点还要询问邻居,节点根据邻居状态和自身的状态确定是事件发生还是自身出错,即第 2 环节。第 3 环节为节点可信度调整环节。可信度指节点检测到的数据可靠程度,可信度越大的节点对邻居决策贡献越大,可信度越小的节点对邻居决策贡献越小。可信度调整的原则为:若节点正确地检测到了事件,就增加其可信度,否则就减小其可信度。

1.1 节点自身检测

通过先验学习,抽象出检测区域中感兴趣的事件的主要的事件属性—时间特性。当事件发生时,将节点采样值变化率与样本时间特性进行匹配。假设 n 个传感器节点均匀分布在感兴趣的区域,用于检测某个特定事件。特定事件的时间序列为 $\{s(\Delta T),s(2\Delta T),\cdots,s(k)\}$,$k=\Delta T,2\Delta T,\cdots,n\Delta T$ 时间序列在 k 时刻的数据变化率(斜率)记为 $r(k)$,则 $r(k)=(s(k)-s(k-\Delta T))/\Delta T$。根据先验经验,$r(k)$ 作为事件的时间特性通常是可以预知的,且在节点部署前已保存在节点的内存中。

节点以 ΔT 为时间间隔进行均匀采样,并建立缓存队列保存当前采样值 $s(k)$ 和前一时刻 $s(k-\Delta T)$。当 $s(k)$ 超过阈值 S_{th},说明特定事件发生或节点发生错误,这时节点启动事件检测程序,并记当前时刻为 $k=0$。

当 $s(k)$ 超过阈值 S_{th},节点启动事件检测程序。如果是事件发生且节点可靠,那么节点采样值在 k 时刻的变化率与 $r(k)$ 应该相差不大,即满足式(1)。但由于节点的不可靠性,正常节点采样值的变化率在少数情况下无法满足式(1),而故障节点采样值的变化率少数情况下能够满足式(1)。

$$\left|\frac{s(k)-s(k-\Delta T)}{\Delta T}-r(k)\right|\leqslant\delta \tag{1}$$

为减小以上情况的影响,本文引入雷达目标检测

中 C_{th}/m 的检测思想[10],即节点连续检测 m 次,若节点采样值每次都大于 S_{th} 且变化率满足式(1)的次数(记为 C)超过 C_{th},那么节点初步认为自己检测到了事件,否则节点初步认为自己发生了错误。为方便描述,记正常状态下的节点状态为 Normal,记初步检测到事件的节点状态为 SE,记初步发生错误的节点状态为 SF。节点自身检测算法描述见算法 1。

算法 1:节点自身检测

输入:节点采样值;输出:节点初步检测状态

```
% for each node
status=Normal;C=0;
while(s(·)≤S_th);% do nothing until sample exceeds threshold
% start event detection procedure
BufferLast=s(·);% create buffer
k=0;
% phase one:local detection
while(every ΔT time)do
    k=k+1;
    Buffer Current=s(k);
    if s(k)>S_th && |(BufferCurrent-BufferLast)/ΔT-r(k)|≤δ
        C=C+1;
    end
    BufferLast=BufferCurrent;
    if k==m
        break;
    end
end
if C≥C_th
    status=SE;
else
    status=SF;
end
```

1.2　事件决策

节点自身检测是节点的自身行为,而自身检测并不能将固有错误消除,其检测结果仍存在着不可靠性。由于通常事件都会覆盖一定的区域和多数节点,于是节点可以通过询问邻居状态来确认事件是否真实发生。邻居节点是指某一节点检测范围内的节点的集合,本文假设节点的检测范围是相同的,且不随能量变化而变化。在某一检测区域中,当一个节点接受到另一节点作为邻居节点的请求后,将会拒绝做其他节点的邻居节点。当该节点同时收到两个或两个以上的节点寻求邻居的请求时,节点优先选择与自己距离较近的节点成为邻居,如果存在与两个或两个以上节点的距离相等的情况,那么该节点随机选择一个节点作为其邻居。本文假设检测区域的节点布置较为密集,任一邻居节点集合都有 2 个以上的邻居节点。

邻居节点在返回的数据包中会包含其自身检测状态和数据可信度。节点收到这些数据包后,做以下运算:

$$W=\sum_{i=1}^{N}\varepsilon_i\cdot CR(i) \tag{2}$$

其中,N 为节点邻居的个数;ε_i 反映了邻居的状态,如果邻居为 SE 状态,那么 ε_i 取 1,如果为 SF 状态就取 0;$\mathrm{CR}(i)$ 为邻居节点 i 的可信度,CR_{self} 为节点自身可信度。处理完邻居的信息,节点得到处理值 W。

根据节点间协作检测的原则[11],如果节点自身状态为 SE 且 $W>\mathrm{CR}_{self}\cdot N/2$,那么节点认为自己正确检测到了事件,状态转移为 Event;否则认为自己发生了虚警错误,状态转移为 Fault。如果节点自身状态为 SF 且 $W>\mathrm{CR}_{self}\cdot N/2$,那么节点认为自己发生了漏警错误,状态转移为 Fault;否则认为没有事件发生,状态转移为 Normal。检测到事件的节点随着事件的消亡也会逐渐恢复正常状态,检测偶尔出错的节点能够回归正常状态,而永久错误节点会被淘汰或弃用。节点状态转移如图 1 所示。

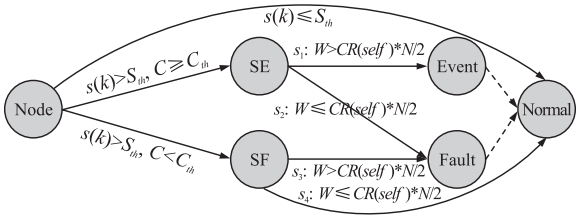


图 1　节点状态转移图

1.3　节点可信度调整

可信度模型是指局部网络区域中节点与邻居节点间可信度的建立、更新和整合的一套机制,根据可信度的分布状况确定节点所处的检测状态,然后根据节点检测状态和能量状态更新节点的可信度。对于正确检测到事件的节点,增加其可信度;对于发生虚警错误和漏警错误的节点,减小其可信度。同时,随着节点能量的消耗,其可信度值也随着下降。可信度的计算遵照式(3)和式(4):

$$\mathrm{CR}(i)=\psi E_c/E_{\mathrm{int}}e^{-y_{\mathrm{curr}}} \tag{3}$$

$$y_{\mathrm{curr}}=\begin{cases}0 & s_1=\mathrm{TRUE}\ \text{and}\ y_{\mathrm{last}}\leq q\\ y_{\mathrm{last}}-q & s_1=\mathrm{TRUE}\ \text{and}\ y_{\mathrm{last}}>q\\ y_{\mathrm{last}}+1-q & s_2=\mathrm{TRUE}\ \text{or}\ s_3=\mathrm{TRUE}\\ y_{\mathrm{last}} & s_4=\mathrm{TRUE}\end{cases} \tag{4}$$

其中,y_{curr} 为节点的故障函数,其最初值为 0,y_{last} 为 y_{curr} 的上一步的值;q 为节点平均出错概率,也即故障节点在所有节点中所占的比例;$\mathrm{CR}(i)$ 的取值范围为(0,1);ψ 为归一化系数;E_c/E_{int} 为节点的能量状态,E_c 当前剩余能量,E_{int} 初始能量;s_1,s_2,s_3,s_4 分

别表示图 1 中状态判定的逻辑变量,TURE 表示状态发生。

可见,如果节点持续错误,其可信度将一直减小;如果节点发生偶然错误,其可信度将很快恢复正常值。同时,随着能量的消耗,节点的可信度也会降低。当 CR_{self}的值低于阈值 ε_{error}时,说明节点的能量消耗过大或节点经常发生错误,我们认为该节点为无效节点,节点从此不再参与事件检测。

节点事件决策和可信度调整的算法描述见算法 2。

算法 2:事件决策和可信度调整

```
输入:节点初步检测状态;输出:事件
W=0;y=0;
% phase two:query neighbor
query and receive neighbor's status and credit…
for i=1:N
    if neighbor's status==SE
        W=W+e^{-y};% y presents neighbor's credit
    end
end
% phase three:make decision and adjust credit
if status==SE
    if W>CR_self·N/2
        status=Event;% event is detected
        if y_last≤q % y_last presents neighbor's last step credit
            y=0;% y presents node's own credit
        else
            y=y_last-q;
        end
    else
        status=Fault;% false alarm
        y=y_last+1-q;
    end
end
if status==SF
    if W>CR_self·N/2
        status=Fault;% missed alarm
        y=y_last+1-q;
    else
        status=Normal;% no event happens
    end
end
```

2 算法性能分析及参数讨论

由算法 1 可以看出,节点在自身检测中会出现两类错误。第 1 类错误为将检测到事件的节点认定为故障节点,即伪故障错误;第 2 类错误为将故障节点认定为检测到事件的节点,即伪正常错误。通常我们都希望这两类错误越小越好。

产生第 1 类错误的原因是,在 m 次连续采样中,节点采样值变化率满足式(1)的次数未达到 C_{th} 次,即至多有 $C_{th}-1$ 次满足式(1)。如果记

$$p=P\left\{\left|\frac{s(k)-s(k-\Delta T)}{\Delta T}-r(k)\right|\leqslant\delta\right\}$$

那么第 1 类错误可以表示为

$$p_1=C_m^i\cdot p^i\cdot(1-p)^{m-i},1\leqslant i\leqslant C_{th}-1,i\in N^* \quad (5)$$

结论 1:当节点通过自身检测的采样值斜率与 $r(k)$匹配次数的阈值 C_{th}越小,同时节点与 $r(k)$相匹配的区分度 δ 越大,伪故障错误发生的概率越小。

产生第 2 类错误的原因是,在 m 次连续采样中,节点采样值变化率满足式(1)的次数超过 C_{th} 次。如果记故障节点的采样值为 $s'(k)$,并记

$$p'=P\left\{\left|\frac{s'(k)-s'(k-\Delta T)}{\Delta T}-r(k)\right|\leqslant\delta\right\}$$

那么第 2 错误可以表示为

$$p_2=C_m^i\cdot(p')^i\cdot(1-p')^{m-i},C_{th}\leqslant i\leqslant m,i\in N^* \quad (6)$$

结论 2:当节点通过自身检测的采样值斜率与 $r(k)$匹配次数的阈值 C_{th}越大,同时节点与 $r(k)$相匹配的区分度 δ 越小,伪正常错误发生概率越小。

由结论 1 和结论 2 可以看出,伪故障错误与伪正常错误是互斥的。由于通常正常节点比故障节点多,因此应该优先考虑降低伪故障错误。

C_{th}和 δ 的取值与实际需求和 $s(k)$服从的分布有关。$s(k)$作为节点的采样值,一般与 $s_m(k)$偏离不大,可以认为 $s(k)$服从以 $s_m(k)$为均值、以 σ^2 为方差的正态分布,同理 $s(k-\Delta T)$服从 $N(s_m(k-\Delta T),\sigma^2)$。由于 $s(k)$与 $s(k-\Delta T)$的均值不同,可以认为 $s(k)$与 $s(k-\Delta T)$相互独立。如果记

$$X=\frac{s(k)-s(k-\Delta T)}{\Delta T}-r(k)$$

那么 X 服从 $N(0,(\sqrt{2}\sigma/\Delta T)^2)$,从而

$$p=P\{|x|\leqslant\delta\}=2\phi(\delta/(\sqrt{2}\sigma))-1 \quad (7)$$

其中 $\varphi(\delta/(\sqrt{2}\sigma))$为标准正态分布。

结论 3:当给定伪故障错误概率 θ,即 $p_1<\theta$,根据式(5)就可以求出 C_{th}和 p,进而解得 δ。

p_2 的取值与 $s'(k)$服从的分布有关。由于故障节点在功能上发生了问题,其产生的采样值是随机的,因此可以认为故障节点的采样值服从[a,b]上的均匀分布,且所有的采样值相互独立。如果记

$$Z=\frac{s'(k)-s'(k-\Delta T)}{\Delta T}-r(k)$$

那么可以得到 Z 的概率密度函数为

$$f_Z(z)=\begin{cases}\dfrac{1}{(b-a)^2}[(\Delta T)^2 z+\Delta T(b-a)+(\Delta T)^2\cdot r(k)] & \dfrac{a-b}{\Delta T}\leqslant z\leqslant -r(k)\\ \dfrac{1}{(b-a)^2}[-(\Delta T)^2 z+\Delta T(b-a)-(\Delta T)^2\cdot r(k)] & -r(k)\leqslant z\leqslant \dfrac{b-a}{\Delta T}-r(k)\\ 0 & \text{others}\end{cases}$$

进一步计算得到

$$p'=P\{|z|\leqslant\delta\}=\frac{1}{(b-a)^2}[2\Delta T\cdot\delta(b-a)-(\Delta T)^2(\delta^2-r^2(k))] \tag{8}$$

p'在区间$(0,+\infty)$上先单调递增后单调递减,在$\Delta T\cdot\delta+(\Delta T\cdot r^2(k))/\delta$处取得最大值。

结论 4:如果要求伪正确错误的概率越小,$b-a$就应该越大。当取定δ后,可以算出p',再结合C_{th}就可以得到p_2。

3　仿真实验

3.1　实验参数

假设 1 024 个节点网格化部署在 32 m×32 m 的区域中,节点通信半径为$\sqrt{2}$ m,事件区域为以(18.5,20.5)中心、半径为 8 m 的圆形区域,如图 2 所示。图中虚线交叉点为正常节点,实心点为故障节点,圆圈所示区域表示事件区域。

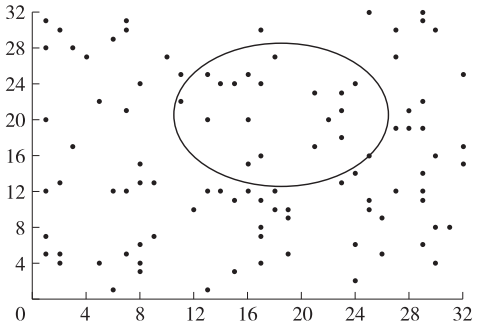


图 2　一次实验快照(故障节点率为 0.1)

假设节点采样频率$\Delta T=1$,$s(k)$服从的正态分布的方差$\sigma^2=1$。如果要求第 1 类错误的概率小于 0.05,那么意味着p_1的最大值要小于 0.05。由于事件发生后大部分节点能以较大概率检测到事件,即p较大(接近 1),此时p_1为i的增函数,因此当$i=C_{\text{th}}-1$时,p_1取最大值。如果节点每采样 10 次做一次自身检测,即$m=10$,要使$p_1<0.05$,那么只要$C_{10}^{C_{\text{th}}-1}\cdot p^{C_{\text{th}}-1}\cdot(1-p)^{11-C_{\text{th}}}<0.05$即可,解得$0.91\leqslant p\leqslant 0.96$和$1\leqslant C_{\text{th}}\leqslant 8$。本文中取$p=0.95$,$C_{\text{th}}=8$。依据式(7),进一步解得$\delta=2.771\ 4$。

假设模式序列的变化率$r(k)=1$。为有效降低第 2 类错误,取$a=S_{\text{th}}$,$b-a=10$,即故障节点服从均匀分布$U(S_{\text{th}},S_{\text{th}}+10)$。为保证故障节点快速得到排除,取节点可信度阈值$\varepsilon_{\text{error}}=0.3$。

3.2　实验与结果分析

为验证算法性能,下面进行 3 组实验。

第 1 组实验考察节点在空间上的协同对漏警率、故障识别率和虚警率的影响。漏警率定义为事件发生时未检测到事件的节点的数目与事件区域总节点数目的比值,故障识别率定义为最终状态为 Fault 的故障节点的数目与事件区域内总故障节点数目的比值,虚警率定义为事件未发生时虚称检测到事件的节点的数目与事件区域总节点数目的比值。

从图 3 可以看到,当节点故障率低于 0.1 时,基于时间和基于时空的方法的漏警率都低于 0.08,即事件检测概率都大于 0.92。但随着节点故障率的增加,利用时空相关性进行检测时的漏警率偏大,这是因为初步检测到事件的节点受到邻居未检测到事件节点的影响,尤其是在节点故障率较大的情况下,初步检测到事件的节点容易转换为 Fault 状态。

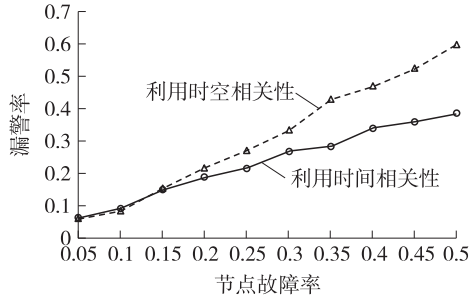


图 3　节点故障率与漏警率的关系

从图 4 可以看出,当节点故障率低于 0.2 时,两种方法的故障节点识别率都能达到 0.93 以上,且性能相差不大,这是因为大部分故障节点通过节点自身检测被滤除掉了。但随着故障率的增加,基于时空相关性的事件检测方法性能下降,这是因为故障节点受到邻居未检测到事件节点的影响,误认为没有事件发生,而转换为 Normal 状态了。

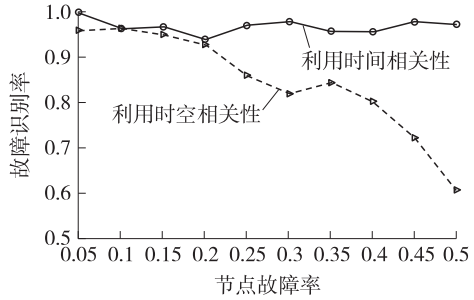


图 4　节点故障率与故障识别率的关系

从图 5 可以看出,利用时空相关性进行事件检测的虚警率远远低于利用时间相关性的,这是因为虚称有事件发生的节点被邻居节点纠正了。

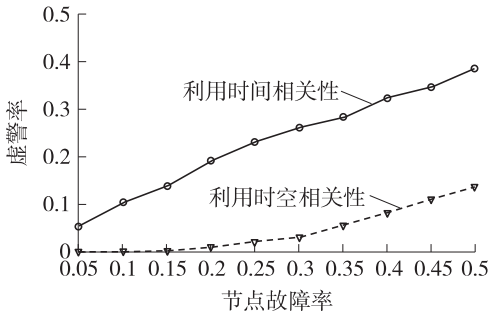


图 5 节点故障率与虚警率的关系

综合以上可以看出,当节点故障率较小(<0.15)时,两种方法在漏警率和故障识别率上相差不大,但由于基于时空相关性的事件检测方法在虚警率上远胜于基于时间相关性的,因此基于时空相关性的事件检测方法略胜一筹。但当节点故障率较大(>0.15)时,就需要根据实际需求进行选择了。由于虚警率和漏警率难以同时达到最优,此时若用户更关注漏警率,那么可以选择基于时间相关性的事件检测方法;若更关注虚警率,那么可以选择基于时空相关性的事件检测方法。

第 2 组实验考察引入节点可信度机制对事件检测概率的影响。仿真中,设定节点故障率为 0.1。

由图 6 可见,引入节点可信度机制的事件检测方法在事件检测概率上一直保持在 90% 左右;而未引入节点可信度机制的事件检测方法,其事件检测概率在第 6 次迭代后就开始下降,最后趋近于零。这是因为前者方法在迭代过程中将无效节点不断的排除掉了,由于剩下的节点都是有效节点,所以事件检测概率能保持较高的值。排除掉的节点与故障节点的关系如图 7 所示。

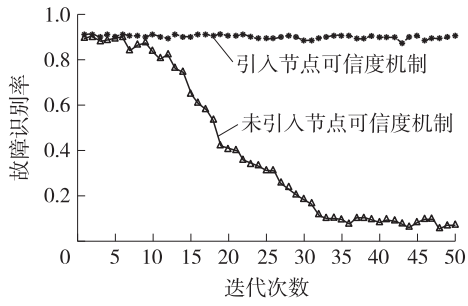


图 6 引入节点可信度机制与未引入的比较

由图 7 可见,在前 19 次迭代过程中,事件区域内剩余有效节点比例曲线下降比较快,而且排除掉的故障节点占排除掉的节点达 70% 以上的比例,这说明大部分故障节点都得到了排除。第 19 次迭代后,由于故障节点所占比例越来越小,事件区域内剩余有效节点比例曲线下降变缓。

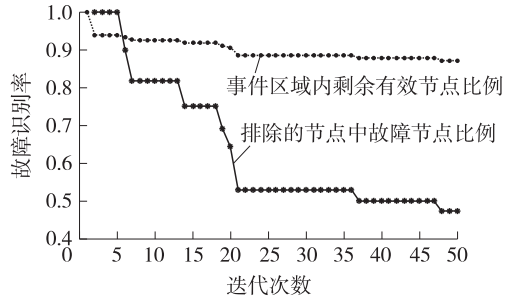


图 7 故障节点与无效节点的关系

第 3 组实验考察网络能耗。网络能耗与节点个数 n、信息交换频率 f、邻居节点个数 N 和数据序列长度 L 有关。传统算法中,节点采集到数据后就与邻居交换进行事件判断,因此能量消耗为 $n\cdot f\cdot N\cdot L$;有些算法在数据交换上采用了数据压缩的策略[12],其能量消耗为 $n\cdot f\cdot N\cdot(\alpha L)$,$0<\alpha<1$;本文中节点与邻居交换的是决策后的信息,即二元决策,其数据长度为 1,且本文数据交换频率为 f/m(m 为时间特性匹配的次数),因此本文方法的能量消耗为 $n\cdot(f/m)\cdot N\cdot 1$。从图 8 对比可见,本文的方法是高能效的。

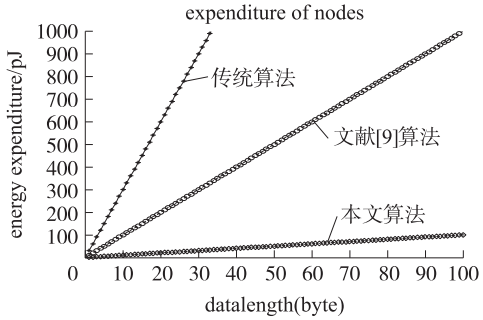


图 8 不同算法节点能量消耗对比图

4 结论

本文讨论了无线传感器网络下一种分布式的轻量级的事件检测方法,该方法充分利用了节点在时间和空间上的相关性。当节点故障率较小的时候,该方法能够获得较低漏警率和虚警率以及较高的故障识别率。由于引入了节点可信度机制,通过不断排除故障节点,该方法能够一直保持较高的事件检测概率。仿真还表明,本文提出的算法是高能效的。

参考文献:

[1] Yick J, Mukherjee B, Ghosal D. Wireless Sensor Network Survey [J]. Computer Networks, 2008, 52(12): 2292-2330.

[2] 张书奎,崔志明,樊建席,等. 基于伸展树的无线传感器网络事件区域检测[J]. 电子学报, 2010, 38(B02): 194-201.

[3] Nguyen-Thanh N, Koo I. Empirical Distribution-Based Event Detection in Wireless Sensor Networks: An Approach Based on Evidence Theory [J]. Sensors Journal, IEEE, 2012, 12 (6): 2222-2228.

[4] Ding M, Chen D, Xing K, et al. Localized Fault-Tolerant Event Boundary Detection in Sensor Networks[C]//INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE. IEEE, 2005, 2: 902-913.

[5] Ould-Ahmed-Vall E, Heck Ferri B, Riley G. Distributed Fault-Tolerance for Event Detection Using Heterogeneous Wireless Sensor Networks[J]. Mobile Computing, IEEE Transaction on, 2012, 11 (12): 1994-2007.

[6] Wittenburg G, Dziengel N, Wartenburger C, et al. A System for Distributed Event Detection in Wireless Sensor Networks [C]// Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks. ACM, 2010: 94-104.

[7] Yim S J, Choi Y H. An Adaptive Fault-Tolerant Event Detection Scheme for Wireless Sensor Networks[J]. Sensors, 2010, 10(3): 2332-2347.

[8] 黄日茂,邱雪松,高志鹏,等. 无线传感器网络中邻居数据分析的故障检测方法[J]. 北京邮电大学学报,2011,34(3):31-34.

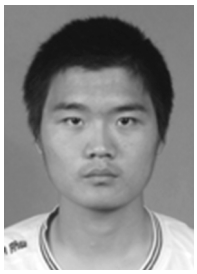
[9] 朱艺华,沈丹丹,吴万登,等. 无线传感器网络优化生存时间的动态路由算法[J]. 电子学报,2009,37(5):1041-1045.

[10] 李建中,李金宝,石胜飞. 传感器网络及其数据管理的概念,问题与进展[J]. 软件学报,2003,14(10):1717-1727.

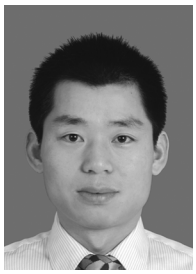
[11] 许小剑,黄培康. 雷达系统及其信息处理[M]. 电子工业出版社,2010.

[12] Krishnamachari B, Iyengar S. Distributed Bayesian Algorithms for Fault-Tolerant Event Region Detection in Wireless Sensor Networks [J]. IEEE Transactions on Computers, 2004, 53(3): 241-250.

[13] 张伟. 无线传感器网络中复杂事件检测关键技术的研究[D]. 哈尔滨:哈尔滨工业大学,2012.



董传明(1987-),男,山东菏泽人,武汉理工大学硕士研究生,主要研究方向为无线传感器网络和交通信息工程;



刘克中(1975-),男,湖北石首人,教授,博士,中国海事科技专家委员会信息分委会委员,主要研究方向为普适计算、无线传感器网络和交通信息工程。