WSN中基于可信核心树的路由算法研究

王江涛 陈志刚 邓晓衡
(中南大学信息科学与工程学院 长沙 410083)

摘 要 介绍了一种适用于WSN网络的基于可信核心树的路由算法。该算法在局部范围内选取具有相对较高剩余能量和可信度的节点作为簇首节点,从而对整个网络进行分簇。然后通过一个最小能耗生成树算法将所有簇首节点组织成以Sink节点为根的可信核心树,进而对该树进行扩展,构建成覆盖全网的可信路由树。所有的数据沿着可信路由树上唯一的路径向Sink节点汇报。同时算法引入了可信模型,该模型通过检测网络中有数据包篡改、丢包、谎报等行为的恶意节点来评估节点的可信度。实验结果证明,该TCTR算法能有效减小能耗、平衡负载,从而延长网络生存时间,同时能有效识别与隔离恶意节点,从而提高路径安全度与网络安全度。

关键词 无线传感器网络,路由树,恶意节点,可信模型
中图法分类号 TP393 **文献标识码** A

Novel Routing Algorithm Based on Trustworthy Core Tree in WSN

WANG Jiang-tao CHEN Zhi-gang DENG Xiao-heng
(School of Information Science and Engineering, Cental South University, Changsha 410083, China)

Abstract A novel routing algorithm based on trustworthy core tree(TCTR) in WSN was proposed in this paper. It aims to prolong network lifetime as well as increase network security in a hierarchical-cluster sensornet. Cluster heads with higher residual energy and trust level were elected from underlying sensor nodes. A minimum pathloss tree algorithm was borrowed to organize all cluster heads as a trustworthy core tree with sink node as tree root. Expanded the Trustworthy core tree to cover all nodes so that each node reports to sink node with a certain route. A trust model was integrated in TCTR to evaluate node's trust level and detect evil nodes. Simulation results testified the effectiveness of the algorithm in producing a longer network lifetime and a safer network.

Keywords WSN, Routing tree, Evil node, Trust model

1 引言

无线传感器网络是一个由大量具有感应和数据处理能力的微型传感器节点组成的自治系统。近年来,无线传感器网络(Wireless Sensor Network,WSN)在多个领域如地震感应、环境监测以及某些军事应用得到了广泛的应用。但由于它是自组织无源网络,无法在布设后补充节点能量,因此其性能很大程度上受限于节点初始能量。如何通过合理的路由策略,在保证数据传输的前提下,尽量节省能量,延长网络寿命,是无线传感器网络中一个需要研究和关注的问题。

同时,随着WSN在军事领域等对安全性要求较高的领域的应用,网络的安全性和可信问题也逐渐成为制约无线传感器网络进一步应用的一个瓶颈,但从事这方面研究的文献相对较少。针对WSN网络的攻击,包括某些被控制的恶意节点,不仅仅通过修改所传输数据或者破坏来达到攻击网络的目的,它还能通过发送恶意的数据来欺骗网络,或通过发送大量的无效数据来浪费网络资源以及恶意地丢弃有效数据包。因此,在无线传感器网络中,如何检测这些恶意节点谎报、丢包、篡改感知数据的行为,从而简单高效地鉴别并隔离恶意节点,提高网络可信度[1,2],增强网络安全性,也是一个值得进一步研究的问题。

WSN网络的体系结构可以分为两大类:基于平面路由和基于分层路由。其中分层路由算法具有能耗小、选路机制简单、扩展方便等有利因素,从而使数据能够得到更加高效、快捷、可靠的传输,适合无线传感器网络应用。近年来,针对分层路由的研究涌现了大量的文献,然而大部分文献都在簇首节点之间采用直接通信的方式来转发数据。研究指出,在WSN中数据传送所耗能量与传送距离的三次方成正比,这种簇首节点间的远距离直接通信会消耗发送节点的大量能量。因此,本文考虑采用一个最小能耗生成树算法将所有的可信簇首节点组织成树形结构,同时在簇首之间采用普通节点作为数据包传送的中转节点,从而节省整个网络的能量。

到稿日期:2011-01-26 返修日期:2011-04-15 本文受国家自然科学基金项目(60873082,60903058),湖南省教育厅科学研究项目(10C0416),湖南省科技计划项目资助。

王江涛(1977-),男,博士生,讲师,主要研究方向为可信计算、无线网络,E-mail:wjt77@163.com;陈志刚(1964-),男,博士,教授,博士生导师,CCF理事,主要研究方向为网络计算与分布式处理、计算机网络;邓晓衡(1974-),男,博士,副教授,CCF高级会员,主要研究方向为计算机网络服务质量、无线网络、网格计算。

目前已经有一些文献分别探讨了无线传感器网络的节能与可信问题。例如在文献[3]中介绍了一种基于熵的可信模型在 WSN 中的应用，该模型可以检测恶意节点，提高网络的安全性；文献[4]研究了一种 WSN 网络中的信任传播机制；文献[5]利用 Beta 分布来描述两个节点之间的可信模型；文献[6]给出了一种基于可信的路由协议 TRANS(Trust Routing for Location-Aware Sensor Networks)；文献[7]利用节点之间的相互观点(opinion)来描述节点的可信问题。R. Kannan 和 S. Iyengar 等在文献[8,9]中均采用了博弈论分别在平面路由和分层路由情况下设计了综合考虑能耗与路径长度的路由协议，却没有考虑节点与路径的可信问题。

以上这些方案都没有综合考虑节点能量约束和节点可信度，因此不能很好地解决 WSN 中的节能与可信问题。综上所述，WSN 中的分层路由算法存在以下几个问题：

1)大部分分层路由算法均采用簇首节点直接长距离与 Sink 节点通信，或者在簇首节点间采用较长距离的直接通信，把数据包层层转发给 Sink 节点。由于数据传送所消耗能量与传送距离的三次方成正比，节点会因能量被大量消耗而“死”掉，网络寿命较短。

2)选择簇首节点时没有考虑节点的可信问题。

3)簇首间的路由选择没有过多考虑能耗问题。

4)没有合适的算法来检测和隔离 WSN 中的恶意节点，网络和路由的安全性没有基本保障。

针对以上问题，本文提出了一种基于可信核心树的路由算法(Trustworthy Core Tree Routing，TCTR)。该算法采用分层路由机制，对网络进行分簇，选取剩余能量和可信度相对较高的节点为簇首节点。然后以网络中的基站 Sink 节点为树根，将网络中的所有簇首节点通过一定的算法组织成可信核心路由树，进而将该可信核心路由树扩建成一棵包括了网络中所有节点的可信路由树。所有的节点都沿唯一的路由来汇报感应数据。同时通过周期性地重新构建该可信路由树来平衡网络负载。本算法综合考虑了路由节能和负载平衡以及节点的可信度问题，以选举出来的具有相对较高剩余能量和较高可信度的簇首节点为骨干构建出覆盖全网的可信路由树。同时在 WSN 网络中实现了可信模型的构建与维护，对恶意节点的数据篡改、谎报、丢包行为进行了惩罚。实验结果显示，本算法能延长无线传感器网络的使用寿命，有效隔离恶意节点，增强路径与网络的安全性。本文的主要创新点如下：

1)通过一个最小路由能耗生成树算法将网络中的簇首节点组织成以 Sink 节点为树根的可信核心树，简化了路由机制。

2)进而在直接相连的簇首节点间选取一个具有最小路由能耗的桥梁节点来中转数据包，对该可信核心树进行扩展，构建了包括网络中所有节点的可信路由树，避免了簇首节点间的直接长距离通信，节约了网络能量，使得所有节点都沿着一条固定的相对能耗小、可信度高的路径汇报数据。

3)在 WSN 中构建与实现了可信模型，借助传统的加密签名技术对恶意节点的数据篡改、谎报、丢包行为进行了惩罚，有效隔离了恶意节点，提高了路由和网络的安全性。

2 相关网络结构描述

本文所提出的 TCTR 路由算法采用分层网络拓扑结构，算法的执行以 round 为时间单位，每个 round 周期由可信路由树构建阶段与数据传输阶段两部分组成。在可信路由树构建阶段，首先所有节点通过设定机制选择剩余能量和可信度相对较高的节点为簇首。当所有节点都有归属簇后，以 Sink 节点为树根将所有簇首节点组织成可信核心树，其次对该可信核心树进行扩展，从而构建覆盖所有节点的可信路由树。在数据传输阶段采用 TDMA 方式，簇首为每个归属节点分配一定的时隙来汇报数据，每个节点沿着可信路由树的唯一路径开始进行数据传输。为了平衡负载，使节点能量消耗分布均匀，在一定时间间隔后重复初始的可信路由树构建过程。

2.1 网络模型假设

(1)假设节点部署完后，通过某个定位算法使每个节点都知道其他所有节点和 Sink 基站的具体位置(因篇幅原因，本文没有给出具体的定位算法)；

(2)Sink 节点是网络中唯一连接电源和因特网的节点，Sink 节点的能量和计算能力以及安全性均不用考虑(即假设 Sink 节点能量无穷且绝对安全)；

(3)所有节点都是平等的(除了 Sink 节点)，节点初始能量可以不同；

(4)所有节点的初始可信度均为 1；

(5)所有节点都有各自的公钥和私钥来对所传输数据进行加密或签名；

(6)网络中的恶意节点以一定的概率 η 进行谎报、篡改、丢包等恶意行为；

(7)节点传输射频范围可调，同时系统存在一个固定的短距离射频范围，用来进行分簇和簇首选举。

3 可信路由树算法(TCTR)

本文提出的可信路由树算法用于在每轮的第一阶段构建覆盖全网的可信路由树。该算法的实现包括了以下几个步骤：步骤一为簇首选举阶段；步骤二为可信核心树构建阶段；步骤三对前面步骤所生成的可信核心树进行扩展，构建覆盖全网的可信路由树。下面分 3 小节来分别描述算法的 3 个步骤。3.4 节简单描述每轮第二阶段的数据传输过程。

3.1 簇首选举

网络中的每一个节点通过以下的公式计算出自身的 π_i，该值用于簇首选举。

$$\pi_i = \alpha E_i / E_{\text{init}} + \beta T_i \tag{1}$$

式中，E_i/E_{init} 表示该节点当前剩余能量与初始能量的比值，该值反映了该节点的能量状况；T_i 表示该节点的可信度；α 和 β 分别为这两个值的权重值，且有 $0<\alpha,\beta<1$ 以及 $\alpha+\beta=1$。簇首选举的过程可以用以下几个步骤来描述：

1)每个节点通过一个全网统一的短距离射频向周围的邻居广播自身的 π 值，每个节点将自身的 π 值与接收到的邻居的 π 值做比较，将那些 π 值比自己大的节点添加到该节点的备选簇首节点集合中。

2)系统中那些自身备选簇首节点集合为空的节点(代表该节点在该局部范围内的所有节点中 π 值最大)确定自己为选举出来的簇首节点。这些簇首节点接下来用短距离射频在局部范围内广播自身为簇首的消息。

3)普通节点接收到一个或多个簇首广播消息后，选取其中 π 值最大的簇首节点为自己的簇首，并向该簇首节点汇报

自己为其成员节点。

4)所有的簇首节点将向 Sink 基站汇报自己为簇首节点并同时汇报自己所覆盖的成员节点。

5)Sink 基站在收到所有簇首节点的汇报后,基于存储在本地的全网节点分布图把网络中剩余的、还没有确定簇首节点的普通节点(这些未确定簇首的普通节点是那些超出簇首节点广播范围的普通节点)根据就近原则选择距离最近的簇首节点为自己的簇首。这个过程结束后,已经确定 Sink 基站中全网的分簇情况,至此簇首选举阶段结束。

3.2 可信核心树的构建

该阶段的主要目标是采用一个最小路由能耗生成树算法构建一个覆盖所有簇首节点的、以 Sink 基站为树根的可信核心树(此阶段该可信核心树的构建只是在 Sink 基站中进行,并未涉及真实的网络),构建的算法如下:

1)根据簇首节点与 Sink 基站的距离,由近至远对所有的簇首节点进行排队,将排队的结果保存在一个向量 Vector 中。

2)初始化可信核心树,初始时该树只有一个树根节点,即 Sink 基站。

3)依次取出排好序的 Vector 中每一个簇首节点,对每一个簇首节点执行以下步骤,执行完毕后该可信核心树构建完毕。

4)对于簇首节点 a,依次尝试将 a 节点作为子节点连接到可信核心树中的每一个树节点的所有连接方法。计算每一种连接方法中从 a 节点到树根 Sink 节点路径的路由能耗(路由能耗模拟为路径中每段子路径的距离的立方之和),在所有这些连接方法中选取路由能耗最小的方法进行实际连接。

此最小路由能耗生成树算法的伪代码图如图 1 所示。

```
ClusterHeadVector={all elected cluster head nodes};
ClusterHeadVector. sortbydistance();//minimum to max distance
TrustCoreTree. root=sinknode;
For(int i=0;i<ClusterHeadVector. size();i++){
  TempClusterHead=ClusterHeadVector. get(i);
  TargetTreeNode = TempCusterHead. findTreeNodeWithMinPath-
Loss();
  TempClusterHead. farthernode=TargetTreeNode;
  TrustCoreTree. add(TempClusterHead);
}
```

图 1 最小路由能耗生成树算法伪代码

3.3 扩展的可信路由树

该阶段的主要目标是扩展前面构建的可信核心树,进而构建覆盖全网所有节点的可信路由树。该阶段由两个子阶段组成。第一个子阶段的目标是在可信核心树上直接相连的树节点间选取桥梁节点来中转数据包。目前大部分文献研究中对于簇首节点间的通信多采用直接通信,但由于节点发送数据包的能量消耗与发送距离的三次方至四次方成正比,因此这种直接通信方式能耗较大。为了避免簇首节点间的长距离通信,本算法在可信核心树上直接相连的簇首节点之间选择一个位置合适的、具有最小能耗的普通节点作为桥梁节点来中转数据包,从而节省路由能量消耗。第二个子阶段的目标是将那些至今为止尚未连接到树的普通节点分别连接到其簇首节点,从而构建覆盖全网的可信路由树。

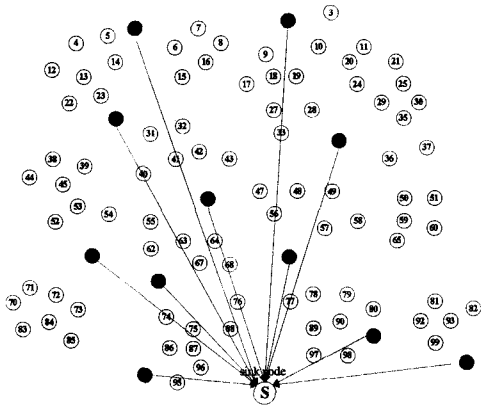

图 2 簇首选举图

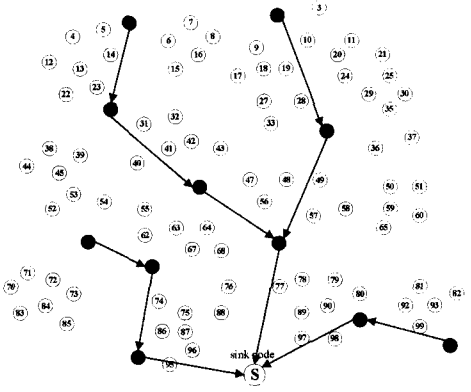

图 3 可信核心树

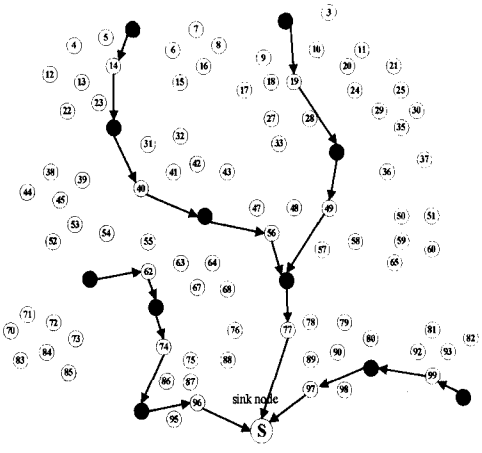

图 4 添加桥梁节点的可信核心树

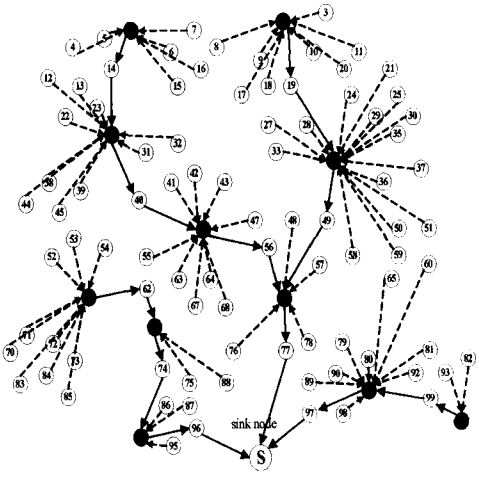

图 5 覆盖全网的可信路由树

此阶段可信路由树的建立只在 Sink 基站进行，并未涉及真实的网络。具体的构建算法如下：

1)对前面建立的可信核心树进行广度优先搜索，搜索过程中对可信核心树的每一条边执行步骤 2)，3)。

2)假设在可信核心树中簇首节点 B 为簇首节点 A 的父节点，那么簇首节点 B 所覆盖的所有成员节点(包括节点 B)组成集合 Set$B$，簇首节点 A 所覆盖的所有成员节点(包括节点 A)组成集合 Set$A$。Set$C$ 为至今为止可信核心树中所有树节点组成的集合。令 Set$D$=Set$A$+Set$B$−Set$C$。

3)在集合 Set$D$ 中选取具有最小中转能耗的节点 K 作为桥梁节点，即具有 $dist^3_{A\to K}+dist^3_{K\to B}$ 最小值并且该最小值小于 $dist^3_{A\to B}$ 的节点为桥梁节点。将原来可信核心树中 $A\to B$ 的边替换为 $A\to K\to B$ 的边。

4)广度优先搜索结束后，对于所有那些已生成的树中未覆盖的普通节点，在已生成的树上添加从该普通节点到其簇首节点的边(即作为其簇首节点的子节点)，至此覆盖全网所有节点的可信路由树构建完毕。

前面 3 小节所描述的可信路由树的构建过程如图 2－图 5 所示。

3.4 路由过程

经过前面几个过程后，所构建的覆盖全网的可信路由树只是建立在 Sink 基站中，并没有涉及真实的网络。接下来 Sink 节点将向全网广播构建好的可信路由树，这样每个节点都接收到该可信路由树，从而每个节点都知道自己向 Sink 基站汇报数据的唯一路径。接下来进入数据传输阶段，在该阶段每个节点感应到事件或者响应某个查询后将根据可信路由树中唯一的路径向 Sink 基站汇报数据，经过一段时间后，为了平衡网络负载，开始重新进入新一轮的可信路由树构建阶段。

4 可信模型

本算法将可信模型引入到了无线传感器网络的路由管理中，目的是借助传统的数据加解密和签名验证技术以及数学统计方法对网络中恶意节点的数据篡改、谎报、丢包等恶意行为进行鉴别和惩罚。因此在本文中可信的概念定义为某个节点行为可信的程度。行为可信亦即该节点没有进行数据篡改、谎报、恶意丢包的行为。一旦某个恶意节点的这些行为被 Sink 节点检测出来，那么该恶意节点的可信度将会因被惩罚而降低。一旦 Sink 节点发现某个节点的可信度低于一个系统设定的恶意节点可信度阈值，那么该节点会因被确认为恶意节点而被加入黑名单隔离起来。

4.1 可信管理

网络中每个节点都有一个可信度，所有的节点可信度值存储在 Sink 节点上，同时每个节点在本地保留一个用于簇首选举的自身可信度值，由 Sink 基站负责维护两个可信度值的一致性。Sink 节点在动态更新某个节点的可信度值后，会通过返回到该节点的 ACK 消息(或者在每轮结束后通过全网广播的方式)来更新存储在节点本地的可信度值，从而维护两个可信度值的一致性。更多可信管理的细节如下所列：

(1)网络初始化时每个节点的可信度都为 1。Sink 节点分析每个节点通过加密技术所发送过来的数据，从而动态评价与更新每个节点的可信度，进而识别出恶意节点，提高网络安全性。

(2)网络中每个节点均有一个公钥私钥对，用来对所发送的数据进行加密或签名，从而可以准确地定位网络中的恶意谎报与篡改以及丢包行为，并有效识别进而孤立恶意节点。

(3)网络中的恶意节点会以一定的概率 η 进行谎报、丢包、数据包修改等等恶意行为。

(4)当网络中的节点监测到事件发生或者要回答查询从而有数据要发送时，该节点先把数据用自己的私钥进行签名，然后把签名后的数据用 Sink 节点的公钥进行加密，再将数据包沿着可信路由树的路径转发，直到 Sink 基站。负责转发的每个中间节点将对收到的数据包进行签名，以保证转发操作的不可否认性，同时便于 Sink 基站的数据分析。

(5)Sink 基站通过解密与验证签名来验证数据源节点与中转节点有没有对数据进行谎报、修改以及丢包。基于本方案的分层网络拓扑结构，使得谎报数据只可能在数据源节点进行，数据篡改和丢包则可能发生在路径上的任意节点上。

4.2 可信度计算

本算法中引入可信模型的设计目标是检测和惩罚有谎报、丢包和修改数据包等行为的恶意节点，降低其可信度，逐渐检测出网络中的恶意节点并及时隔离。Sink 节点分析所有节点的汇报数据后，可以确定节点谎报、篡改数据包、丢包等行为，从而更新 Sink 节点内存储的节点的可信度值。每一轮的数据传输阶段结束后，Sink 基站将统计每个节点的可信度，将那些可信度 R_i 低于系统预设阈值 R_{th} 的节点确认为恶意节点。向全网广播这些恶意节点，从而隔离这些恶意节点，提高网络的可信度和安全性。

几种恶意行为的检测过程和相应的可信度的惩罚公式用以下 3 小节来描述。

4.2.1 数据包篡改行为的检测和惩罚

对于恶意节点的篡改数据包行为，本文设计如下方案进行鉴别和惩罚。如果经过签名验证后 Sink 基站发现某个收到的数据包已经被篡改，则认为篡改行为可能发生在该数据包转发路径上的任意一个或多个节点上。也即如果发生了一次数据包篡改行为，那么 n 个中转节点中每个节点都有可能进行了篡改操作。Sink 基站将认为数据包路径上每个中转节点都进行了一次数据篡改，并对每个中转节点都记录下这一次数据篡改行为。每轮的数据传输阶段结束后，在新的可信路由树构建阶段之前，Sink 基站将统计每个节点在本轮数据传输阶段的数据包篡改次数 q。图 6 为某轮实验结束后，网络中每个节点的数据包篡改次数统计图。图中红色节点为仿真实验前系统随机指定的恶意节点，蓝色数字为对应节点的数据包篡改次数统计。

分析该图可以得到以下结论：

1)如果某个节点(除开 Sink 节点)的数据包篡改次数等于其所有子节点的数据包篡改次数之和，则该节点存在数据包篡改行为的概率较小，本模型认为该节点没有数据包篡改行为。

2)如果某个节点(除开 Sink 节点)的数据包篡改次数大

于其所有子节点的数据包篡改次数之和，则该节点存在数据包篡改行为的概率较大，本模型认为该节点存在恶意数据包篡改的行为，并将通过以下的惩罚公式对该节点的可信度进行惩罚，每轮惩罚结束后数据包篡改次数 q 清零。

$$R_i=(1-\theta)R_i \quad (0\leqslant R_i,\theta\leqslant 1) \tag{2}$$

式中，R_i 为节点 i 的可信度，θ 为数据篡改惩罚系数。

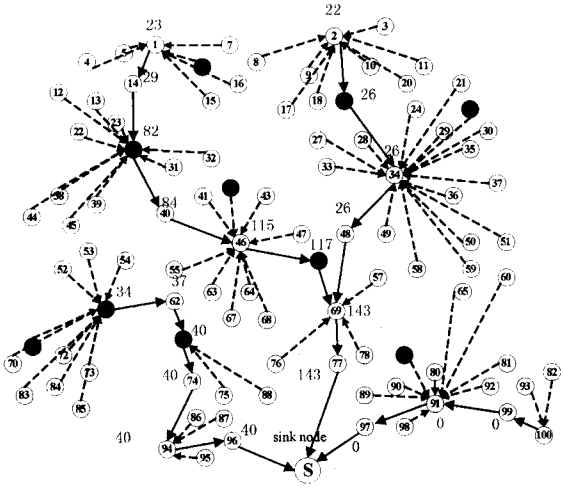

图 6　数据篡改次数统计图

4.2.2　丢包行为的检测和惩罚

本文设计如下方案来鉴别和惩罚恶意节点的丢包行为。本方案假设每次 Sink 节点收到数据后会对数据包做一定的记录，记录下数据发送的源节点和中转节点，也记录数据发送的时间和数据包大小。数据包在中转节点的层层转发过程中，都有一个数据包收到确认过程。如果某个节点 i 在发送数据后超过时限 ϕ 还没有收到下个中转节点 j 的数据确认消息，则认为该数据包已经被节点 j 丢弃，那么该节点 i 将越过节点 j 而直接通过节点 j 的父节点 k 向 Sink 基站报告该次丢包事件。这时 Sink 节点会统计该时间段内节点 j 的数据处理量。如果发现该段时间内节点 j 的数据处理量并没有超过节点的饱和数据处理量 ψ，则认为节点 j 存在恶意丢包行为，将利用以下公式对其进行可信度惩罚。

$$R_j=(1-\mu)R_j \quad (0\leqslant R_j,\mu\leqslant 1) \tag{3}$$

式中，R_j 为节点 j 的可信度，μ 为丢包惩罚系数。

4.2.3　谎报行为的检测与惩罚

针对恶意节点的数据谎报行为，本文设计如下方案（该方案仅对数据查询有效）。假设 Sink 节点发出一次数据查询请求后，会有多个指定的传感器节点被要求汇报数据来响应该查询请求。本方案假设局部小范围内（即簇内）的节点针对同一次查询而汇报数据时，它们所汇报的数据值应该具有一定的相似度，即数据的一致性应该较高。如果某个节点所汇报的数据与簇内其他节点所汇报的数据相对偏差太大，则认为该节点存在数据谎报的行为，从而降低该节点的可信度。具体检测和惩罚方法如下：

Sink 基站接收到节点的汇报数据后，将分簇统计簇内所有节点的汇报数据的加权平均值。即根据各个节点的可信度，对簇内所有的汇报数据求加权平均值。

$$V_w=\frac{\sum_{i=1}^{n}R_iV_i}{\sum_{i=1}^{n}R_i} \tag{4}$$

式中，n 为该簇内汇报事件的节点个数，R_i 为节点 i 的可信度值，V_i 为节点的汇报数据值。则某个节点汇报数据的相对偏差可以用以下公式计算：

$$Diff_i=\frac{V_w-V_i}{nSTD} \tag{5}$$

式中，n 为簇内汇报事件的节点个数，STD 为这 n 个汇报数据的标准偏差，$Diff_i$ 为节点 i 汇报数据的相对偏差，可以用来衡量数据的不一致性。

节点 i 的可信度更新公式如下：

$$R_i^*=\begin{cases}R_i+\lambda(1-R_i)(1-Diff_i), & 0\leqslant Diff_i\leqslant 1\\ R_i-\chi R_i(1-\dfrac{1}{Diff_i}), & Diff_i>1\end{cases} \tag{6}$$

式中，R_i^* 为节点 i 新的可信度，λ 为奖励系数，χ 为谎报惩罚系数。

4.3　可信模型有效性分析

本算法中可信模型的设计目标是检测有谎报、丢包和修改数据包等行为的恶意节点，对其进行惩罚，降低其可信度，逐渐检测出网络中的恶意节点并及时隔离。本模型假设网络中的恶意节点会以一定的概率进行谎报、丢包和篡改等恶意行为，而根据本可信模型这些有恶意行为的节点会被一一检测出来，其可信度也会受到相应的降低惩罚。而在新的一轮的簇首选举阶段，根据本方案的簇首选举方法，那些有恶意行为的节点也会因为其可信度受到惩罚而降低了其被选举为簇首参与路由的概率，也就意味着网络中的恶意节点会被逐渐排斥在网络的边缘，从而减小了其进行恶意行为的机会概率，有效地提高了网络的安全性。后面的仿真实验也验证了本可信模型能有效检测出有恶意行为的节点并逐渐隔离这些恶意节点。

5　仿真实验结果及分析

本算法设计的一个主要目标是节约能耗和平衡负载，以提高网络生存时间。另外一个主要目标是隔离恶意节点，提高路径与网络的安全度。因此在仿真实验中我们采用以下几个指标来衡量算法的性能。

• 网络生存时间：实验中采用两个时间点来衡量算法的网络生存时间性能，分别是首节点死亡轮数和半数节点死亡轮数。相对较高的两个节点死亡轮数可以证明算法能有效节约能耗，从而延长网络寿命。

• 剩余节点能量方差：网络中剩余存活节点的能量方差是衡量负载平衡的有效方法。低的剩余节点能量方差证明算法能有效平衡网络负载，从而延长网络寿命。

• 剩余节点平均可信度：网络中剩余存活节点的平均可信度是衡量 WSN 网络整体可信水平的有效方法。高的剩余节点平均可信度证明该网络整体可信度高，安全性也高。

• 路径安全度：该指标通过以下公式定义：

$$PS_n=\frac{NSP_n}{NRP_n} \tag{7}$$

式中，PS_n 为单轮实验的路径安全度，NSP_n 为该轮实验中 Sink 节点接收到的没有经过谎报和篡改的数据包的个数，NRP_n 为该轮实验中所有源节点所报告的数据包的个数之和。这个指标统计每轮的实验结束后，Sink 节点接收到的安

全数据包占该轮所有节点汇报数据包的比例。该值可以反映路由和网络安全状况。

5.1 相关参数设定

α,β的设定：α,β分别是簇首选举期间计算节点π值的能量权重参数和可信度权重参数，实验中它们分别设置为0.6和0.4。

θ,μ的设定：θ,μ分别是针对数据包篡改行为和恶意丢包行为的可信度惩罚参数，在实验中它们都设置为0.2。

R_{th}的设定：R_{th}是识别恶意节点的可信度阈值，该阈值是一个经验性的参数，可以通过对以往历史的观察来进行更新，R_{th}的设定可以参考如下公式：

$$R_{th\text{-}current}=\sigma R_{th\text{-}prev1}+(1-\sigma)R_{th\text{-}prev2} \tag{8}$$

式中，$R_{th\text{-}prev1}$和$R_{th\text{-}prev2}$分别为上一轮和上上轮的R_{th}值，σ为历史权重参数。

5.2 仿真实验结果及分析

在仿真实验中，100个节点随机分布在1000m×1000m的平面空间中。实验将可信路由树算法TCTR与WSN传统的分层路由算法LEACH进行比较，用网络生存时间和剩余节点能量方差两个指标来衡量两个算法的性能。与此同时，仿真实验通过统计每轮实验的剩余节点平均可信度和路径安全度来衡量TCTR算法中可信模型的有效性。实验中的能量消耗主要考虑节点发送和接收数据包时消耗的能量，其中节点发送数据包所消耗的能量与发送距离的三次方成正比。

5.2.1 网络生存时间比较

这部分实验将比较TCTR算法和LEACH算法的网络生存时间性能，将分别比较两个算法的首节点死亡轮数和半数节点死亡轮数(见图7)。该组实验将执行10次，每次实验分为50轮，每轮实验中将随机地选择500个节点作为源节点来向Sink节点汇报数据。

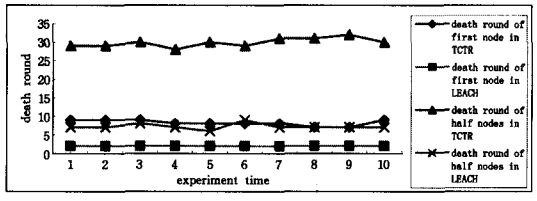

图7 节点死亡轮数统计图

从实验结果中可以看出，TCTR算法在网络生存时间性能上好于LEACH算法，TCTR算法的首节点和半数节点死亡轮数明显高于LEACH算法，可能的原因包括：

- TCTR路由算法采取层层中转的方式把数据汇报给Sink节点，避免了长距离数据传输，节约了大量的能量，这是该算法网络寿命较长的主要原因。
- LEACH算法的簇首分布不均匀，从而导致能量消耗相对集中，一些"热"节点迅速死亡。
- LEACH算法的簇首选举过程中采用了全网广播的方式，这种远距离数据传送消耗了节点大量的能量。

5.2.2 剩余节点能量方差比较

图8显示了每轮实验结束后两种算法的网络剩余节点的能量方差。相对而言，LEACH算法的剩余节点能量方差高于TCTR算法，这说明在LEACH中能耗相对集中，部分"热"节点会因为过度消耗而死亡。从实验结果来看可以说明TCTR算法的能耗分布均匀，这也是其网络生存时间较长的原因之一。

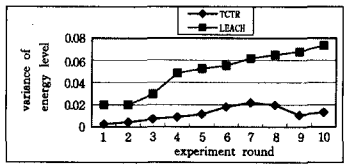

图8 两种算法的剩余节点能量方差比较

5.2.3 TCTR算法剩余节点平均可信度

该部分实验统计了总共20轮实验后网络中每轮剩余存活节点的平均可信度的变化趋势。从实验结果中可以看出，TCTR算法网络中剩余存活节点的平均可信度存在一个先下降后上升的趋势，这是因为初始时节点的可信度初始化均为1，随着实验的进行，恶意节点会随机地进行谎报、修改、丢弃数据等恶意行为，这些行为被Sink节点察觉后会对恶意节点进行降低可信度的惩罚，因而节点平均可信度会有下降的趋势。图9中曲线到了第5轮有一个上升的趋势，这是由于随着Sink节点对恶意行为的惩罚，恶意节点会被慢慢识别出来，从而被排除出整个网络，因此网络剩余存活节点的平均可信度会回升。到了第10轮后，网络中所有恶意节点被排除出网络，所以平均可信度回升到1。从实验结果可以看出TCTR算法可以迅速、简单、有效地识别出恶意节点。

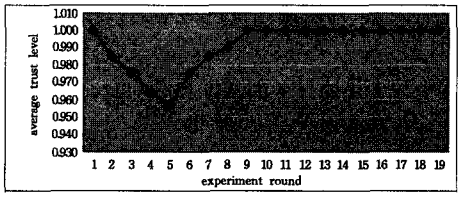

图9 TCTR算法剩余存活节点平均可信度变化图

5.2.4 路径安全度

该部分实验显示了每轮实验结束后该轮实验的路径安全度。实验开始之前，在网络中随机选择10个节点作为恶意节点，这些节点会以一定的概率η来进行谎报、丢包和数据包篡改等恶意行为。Sink节点通过分析会发现这些节点的恶意行为并对其可信度进行惩罚，一旦发现某个节点的可信度低于系统设定的阈值R_{th}，该节点就会被确认为恶意节点而被隔离。

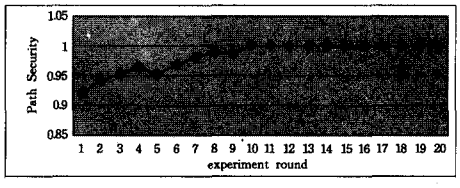

图10 TCTR算法的路径安全度统计图

从图10可以看出，随着实验的逐轮进行，路径安全度PS_n逐渐上升，一直到第10轮后稳定在较高水平。可能的原因是恶意节点逐渐被检测并且隔离起来。实验结果显示TCTR算法能有效识别恶意节点，提高路径和网络安全度。

结束语 本文介绍了一种适用于WSN网络的基于可信核心树的路由算法。该算法在局部范围内选取具有相对较高剩余能量和可信度的节点作为簇首节点，从而对整个网络进行分簇。通过一个最小能耗生成树算法将所有簇首节点组织成以Sink节点为根的可信核心树，进而将该树扩建成覆盖全网的可信路由树。同时，算法引入了可信模型，该模型通过检

测网络中有数据包篡改、丢包、谎报等行为的恶意节点来评估节点的可信度。实验结果证明，该 TCTR 算法能减小能耗、平衡负载，从而延长网络生存时间，同时能快速有效地识别与隔离恶意节点，从而提高路径和网络的安全度。

参 考 文 献

[1] 荆琦，唐礼勇，陈钟. 无线传感器网络中的信任管理[J]. 软件学报，2008，19(7)：1716-1730

[2] 陈钟，刘鹏，刘欣. 可信计算概论[J]. 信息安全与通信保密，2003(11)

[3] Dai Hong-jun, Jia Zhi-ping, Dong Xiao-na. An Entropy-based Trust Modeling and Evaluation for Wireless Sensor Networks [C]// Proceedings of the 2008 International Conference on Embedded Software and Systems. Washington, DC, USA, 2008：27-34

[4] Xiong Fei, Xu Qi-jian. Active Trust Transmission Mechanism for Wireless Sensor Network [C]// Proceedings of the 2008 Second International Symposium on Intelligent Information Technology Application. Washington, DC, USA, 2008：626-632

[5] Ganeriwal S, Srivastava M B. Reputation-based Frameworkfor High Integrity Sensor Networks [C]// SASN '04. Washington, USA, 2004

[6] Tanchaiwiwat S, Dave P, Bhindwale R, et al. Secure Locations：Routing on Trust and Isolating Compromised Sensors in Location-aware Sensor Networks[C]// SenSys '03. California, USA, 2003：324-325

[7] Li Xiao-qi, Liu M R, Liu Jiang-chuan. A TrustModel-based Routing Protocol for Secure Ad Hoc Networks[EB /OL]. http：www. cse. cuhk. edu. hk /～lyu /paper_pdf /Aero04_TAODV. pdf

[8] Kannan R, Ray L, Iyengar S S, et al. Maxmin length-energy-constrained routing in wireless sensor networks[C]// Proceedings of 1st European Conference Workshop on Wireless Sensor Networks. Berlin, Germany, January 2004：234-249

[9] Kannan R, Iyengar S S. Game-theoretic models for reliable path-length and energy-constrained routing with data aggregation in wireless sensor networks[J]. IEEE Trans. on Selected Areas of Communications, 2004, 22(6)：1141-1150

(上接第 30 页)

待进一步研究。虚拟资源供给算法现在大都基于某种假设，还需要对更多约束和限制条件进行考虑，以及从全局角度进行优化设计。

(2)在进行大规模任务的调度和分配时，仍有很多问题亟待解决，如任务依赖关系的处理、节点的动态选择等，需要根据具体任务特性和范围进行优化设计。另外，利用虚拟资源基本元素间的相互关系，需要形成更能均衡成本、服务质量和效率等因素的资源按需调度方法。

(3)在虚拟适配层还可以考虑更多的内容适配，如基于语义的内容适配等。

参 考 文 献

[1] Douze M, Gaidon A, Jegou H, et al. INRIA-LEAR's video copy detection system [C]// Proc. the TRECVID Workshop. 2008

[2] Xu Peng, Xie Le-xing, Chang S F, et al. Algorithms and system for segmentation and structure analysis in soccer video [C]// Proc. the IEEE ICME. 2001：928-931

[3] Gauch J, Shivadas A. Identification of new commercials using repeated video sequence detection [C]// Proc. the IEEE International Conference on Image Processing. 2006

[4] Shen Heng-tao, Zhou Xiao-fang, Huang Zi, et al. Statistical summarization of content features for fast near-duplicate video detection [C]// Proc. the 15th International Conference on Multimedia. 2007：164-165

[5] Zhao Wan-lei, Tan Song, Ngo C W. Large-scale near-duplicate web video search：Challenge and opportunity [C]// Proc. the IEEE International Conference on Multimedia and Expo. 2009：1624-1627

[6] Shang Li-feng, Yang Lin-jun, Wang Fei, et al. Real-time large scale near-duplicate web video retrieval [C]// Proc. the International Conference on Multimedia. 2010：531-540

[7] Hoad T, Zobel J. Fast video matching with signature alignment [C]// Proc. the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval. 2003：262-269

[8] Lejsek H, Tormóesdóttir H, ásmundsson F, et al. Videntifier Forensic：large-scale video identification in practice [C]// Proc. the 2nd ACM Workshop on Multimedia in Forensics, Security and Intelligence. 2010：1-6

[9] Vouk M A. Cloud computing-Issues, research and implementations [J]. Journal of Computing and Information Technology, 2008, 16(4)：235-246

[10] Armbrust M, Fox A, Griffith R, et al. Above the clouds：A berkeley view of cloud computing [R]. UCB/EECS-2009-28. EECS Department, University of California, Berkeley, 2009

[11] Zhu Yong, Ammar M. Algorithms for assigning substrate network resources to virtual network components [C]// Proc. the IEEE INFOCOM. 2006

[12] Yu Min-lan, Yi Y, Rexford J, et al. Rethinking virtual network embedding：Substrate support for path splitting and migration [J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2)：17-29

[13] Lu Jing, Turner J. Efficient mapping of virtual networks onto a shared substrate [R]. WUCSE-2006. Department of Computer Science and Engineering, Washington University in St. Louis, 2006

[14] Courconbetis C, Weber R R. Economic issues in shared infrastructures [C]// Proc. the VISA. 2009：89-95

[15] Heymann E, Senar M, Luque E, et al. Evaluation of an adaptive scheduling strategy for master-worker applications on clusters of workstations [J]. High Performance Computing, 2000：310-319

[16] Dean J, Ghemawat S. Mapreduce：Simplified data processing on large clusters [J]. Communications of the ACM, 2008, 51(1)：107-113

[17] Motahari Nezhad H R, Benatallah B, Martens A, et al. Semi-automated adaptation of service interactions [C]// Proc. the International Conference World Wide Web. 2007：993-1002