

# 延迟容忍移动传感器网络中基于节点优先级的数据转发策略

刘唐<sup>1</sup> 彭舰<sup>2</sup> 王建忠<sup>1</sup> 刘浏<sup>3</sup>

(四川师范大学基础教学学院 成都 610068)<sup>1</sup> (四川大学计算机学院 成都 610065)<sup>2</sup>

(四川师范大学数学与软件学院 成都 610068)<sup>3</sup>

**摘要** 为了更好地解决延迟容忍移动无线传感器网络(Delay Tolerant Mobile Sensor Network, DTMSN)中的数据收集问题,提出了一种基于节点优先级的数据转发策略 NPD(Node Priority Data Delivery Scheme)。NPD 根据计算得出的各节点不同的转发优先级作为消息传输时选择下一跳的依据。为优化消息副本管理,NPD 采用动态消息队列,并根据消息的生存时间决定消息的丢弃原则。仿真实验表明,与现有的几种 DTMSN 数据传输算法相比,NPD 有更高的数据传输成功率与更小的传输延迟,并具有相对较长的网络寿命。

**关键词** 延迟容忍移动无线传感器网络,数据收集,动态数据传输,节点优先级,队列管理

**中图分类号** TP393 **文献标识码** A

## Data Delivery Scheme of Delay Tolerant Mobile Sensor Network Based on Node Priority

LIU Tang<sup>1</sup> PENG Jian<sup>2</sup> WANG Jian-zhong<sup>1</sup> LIU Liu<sup>3</sup>

(College of Fundamental Education, Sichuan Normal University, Chengdu 610068, China)<sup>1</sup>

(College of Computer Science, Sichuan University, Chengdu 610065, China)<sup>2</sup>

(College of Mathematics and Software Science, Sichuan Normal University, Chengdu 610068, China)<sup>3</sup>

**Abstract** To better solve the problem of data collection in DTMSN(delay tolerant mobile sensor network, DTMSN), this paper proposed a new data delivery scheme-NPD(Node Priority Data Delivery Scheme). NPD calculates the node delivery priority, according to which nodes make decisions when delivering messages. To optimize the management of information copy, NPD adopted dynamic message queue and decided on the principle of abandoning messages based on their survival time. Simulation results show that compared with the existing algorithm of DTMSN data transmission, NPD has a higher delivery ratio in data transmission and smaller delivery delay and it has a relatively longer network lifetime.

**Keywords** DTMSN, Data collection, Dynamic data delivery, Node priority, Queue management

## 1 引言

近年来,面向移动对象的无线传感器网络成为了研究热点<sup>[1]</sup>。由于数据收集对象的移动性、随机性必然导致节点间非连通状态的频繁出现,因此节点间数据的传递也会相应地出现数据延迟。

延迟容忍网络 DTN(delay tolerant network)<sup>[2]</sup>用于间歇连通的网路传送数据。文献[3]给出了 DTN 的总体结构,作为网络互联时传输层上的覆盖网,它提供了数据存储和重传、经过鉴别的数据传输等服务。DTN 技术近几年被引入无线传感器网络,并且根据节点移动性的不同,可以把延迟容忍传感器网络 DTSN(delay tolerant sensor networks)分为传感器节点静止的网络和节点随机运动的传感器网络 DTMSN<sup>[4]</sup>。

对于 DTMSN,最关键的是如何设计 DTMSN 环境下的

路由算法。目前已有一些文献对此进行了研究。其中最简单的方法为文献[5]所提出的直接传递(Direct Transmission),基本思想是只有当传感器节点运动到汇聚点的通信范围内时两者才进行消息传输。虽然它的传输能耗非常低,但是消息的传输延迟非常大。文献[6]提出了泛洪算法(Flooding),令传感器节点把消息复制给它的通信范围内的所有节点。这种算法能够以消耗大量传感器能量为代价获得较高的数据传输成功率。然而此算法丢包现象严重,性能较差。文献[7]提出的 RED 策略由数据传输和消息管理两部分组成。通过消息传输的发生与否来增加或降低传输概率。消息管理则根据节点当前的传输概率值确定最佳的容错编码参数以提高传输成功率。然而,文献[5]认为 RED 中最佳容错编码参数的计算值并不准确,并且在网络中传输大量小的分片消息,进一步加剧了网络的传输能耗。文献[8]提出 SRAD 策略,SRAD 根据

到稿日期:2010-04-23 返修日期:2010-07-18 本文受四川省科技支撑项目(2009GZ0153),四川省教育厅科研项目(10ZB005)资助。

刘唐(1980-),男,硕士,讲师,主要研究方向为无线传感器网络、P2P 计算,E-mail:crikey@163.com;彭舰(1970-),男,博士,教授,主要研究方向为并行计算;王建忠(1965-),男,博士,副教授,主要研究方向为光交换网;刘浏(1981-),男,硕士,讲师,主要研究方向为无线传感器网络。

节点运动轨迹与汇聚点的最小距离确定节点的转发概率,并利用消息的生存时间进行消息队列管理。但 SRAD 没有考虑到消息传递距离对发送能量的比值关系,依旧有较高的能效消耗,并且存在较大的消息冗余。与 SRAD 相似,文献[9]提出的 PDR 依然因同样的原因存在较高的能量消耗与消息冗余。

为合理利用网络能量,延长网络的使用寿命,以尽可能低的传输延迟来达到尽可能高的数据传输成功率,本文提出了一种基于节点优先级的数据转发策略 NPD(Node Priority Data Delivery Scheme)。其基本思想是综合节点运动路径、节点剩余能量、节点运动速率等因素计算出节点传输优先级,并在数据转发时根据通信节点间的距离得到节点转发优先级,它是节点进行消息传递时选择下一跳的重要依据。进一步,NPD 建立动态消息队列,并根据消息的生存时间决定消息的丢弃原则。

## 2 网络模型与传输策略

### 2.1 网络模型

本文中  $N$  个传感器节点在初始状态时随机均匀分布在一个正方形区域  $A$  内,并且该传感器网络具有如下性质。

(1)所有传感器节点的运动规律符合 Random Waypoint 运动模型<sup>[8]</sup>。Random Waypoint 运动模型描述为:传感器节点在运动空间  $A$  内随机选取起始点  $S$  和目的点  $D$ ,随机选取  $V$  属于  $(V_{\min}, V_{\max})$  作为此次运动的速度,匀速从  $S$  沿直线运动到  $D$ ,在  $D$  随机选取一个时间  $T_{\text{pause}}$  属于  $(T_{\min}, T_{\max})$  保持静止,这样完成一次运动过程,将本次的目的点  $D$  作为下次运动的起始点  $S$ ,进行下一次运动过程,如此重复。网络中所有传感器节点均遵循上述运动过程,它们之间相互独立。节点的运动过程如图 1 所示。

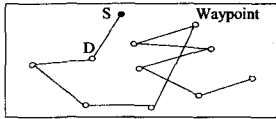


图 1 采用 Random Waypoint 运动模型的节点运动模式

(2)所有汇聚点以基站形式部署,因此不考虑汇聚点能量问题,汇聚点部署后便不再移动。

(3)部署后传感器节点不需要人为维护。

(4)通过全球定位系统 GPS(global positioning system),各个传感器节点可获知任意时刻自己的当前位置及运动速率。

(5)所有传感器节点保持时钟同步。

### 2.2 通信模型

本文采用与文献[10]相同的无线通信模型,该模型给出了一个阈值  $d_0$  ( $d_0$  是常数,数值取决于使用环境),当发送节点与接收节点的距离小于  $d_0$  时,发送方发送数据的能量损耗与距离的平方成正比,否则与距离的四次方成正比。发送节点与接收节点的能量消耗可用下列公式计算:

$$E_{tr}(k, d) = E_{dec}(k) + E_{amp}(k, d) = \begin{cases} kE_{dec} + k\epsilon_{-f}d^2, & d < d_0 \\ kE_{dec} + k\epsilon_{-mp}d^4, & d \geq d_0 \end{cases} \quad (1)$$

$$E_{rc}(k) = kE_{dec} \quad (2)$$

本文假设无线信道是对称的,即从节点  $a$  传送消息  $m$  到

节点  $b$  消耗的能量等于从节点  $b$  传送消息  $m$  到节点  $a$  消耗的能量。

### 2.3 节点传输优先级计算

传感器节点传输优先级作为数据转发时转发对象选择的依据。令  $P_i$  表示节点  $i$  的传输优先级,因为随机运动模型的特点, $P_i$  的值是变化的。

根据本文所使用通信模型,节点  $i$  首先判断自己所在位置,若当前位置在距离汇聚点  $d_0$  范围内,该节点传输优先级  $P_i = 1$ 。若当前位置不在距离汇聚点  $d_0$  范围内,则判断节点  $i$  的当前位置  $S(x_1, y_1)$  和这次运动的目的点  $D(x_2, y_2)$  所成线段是否经过汇聚点  $O$ ,即汇聚点到线段  $SD$  的距离  $l$  是否等于 0,若经过  $O$ ,则  $P_i = 1$ 。设汇聚点位置为原点,计算过程如下:

$$l = \frac{x_1 y_2 - x_2 y_1}{\sqrt{(y_1 - y_2)^2 + (x_2 - x_1)^2}} \quad (3)$$

若节点传输优先级  $P_i$  不为 1,则节点计算出自身  $P_i$ ,并将其作为数据转发的判断依据:

$$P_i = \alpha \bar{\omega}(D_i) \times \beta \bar{\omega}(E_i) \times \gamma \bar{\omega}(V_i) \quad (4)$$

式中, $\bar{\omega}(D_i)$  为距离函数,反映节点  $i$  从当前点  $S$  到目的点运动路径  $D$  所成线段距离汇聚点  $O$  的距离  $l$  对节点传输优先级的影响。根据本文所用通信模型,设汇聚点通信阈值为  $d_0$ ,汇聚点最大通信半径为  $d$ 。当节点  $i$  运动路径距离汇聚点最小距离  $l \rightarrow 0$  时, $\bar{\omega}(D_i) \rightarrow 1$ ;当  $l \rightarrow d$  时, $\bar{\omega}(D_i) \rightarrow 0$ 。充分考虑阈值  $d_0$  对能量消耗的影响,当  $l \in (0, d_0]$  时:

$$\bar{\omega}(D_i) = 1 - \sqrt{\frac{l}{d}} \quad (5)$$

而对于  $l \in (d_0, d]$  时,可令:

$$\bar{\omega}(D_i) = a(l-d)^4 \quad (6)$$

因为  $l \rightarrow d$  时, $\bar{\omega}(D_i) \rightarrow 0$ ,可知:

$$a = 1 - \sqrt{\frac{l}{d}} / (d_0 - d)^4 \quad (7)$$

因此由上述 3 式可得出距离函数  $\bar{\omega}(D_i)$  如下:

$$\bar{\omega}(D_i) = \begin{cases} 1 - \sqrt{\frac{l}{d}}, & l \in (0, d_0] \\ (1 - \sqrt{\frac{l}{d}}) \frac{(l-d)^4}{(d_0-d)^4}, & l \in (d_0, d] \end{cases} \quad (8)$$

$\bar{\omega}(E_i)$  为能量函数,用以调节节点当前剩余能量情况对节点传输优先级的影响:

$$\bar{\omega}(E_i) = \frac{E_{cur}}{E_{mit}} \quad (9)$$

式中, $E_{cur}$  表示节点当前的能量值, $E_{mit}$  表示节点的初始能量值。

$\bar{\omega}(V_i)$  为速率函数。若某节点速率越快,则越能在更短时间内运动到汇聚点通信范围内,从而有更小的延迟,因此考虑对该节点给予更高的传输优先级。节点运动速率为  $v_i$ , $\bar{\omega}(V_i)$  计算如下:

$$\bar{\omega}(V_i) = e^{-\frac{1}{v_i}} \quad (10)$$

$\alpha, \beta, \gamma$  为调节距离函数、能量函数、速率函数在计算节点传输优先级时所占权重的计算因子。

### 2.4 数据转发策略

若节点  $a$  的存储队列头部有一个消息  $j$  正等待传输,数据消息的转发根据节点数据转发优先级  $T_m$  进行。假设目前  $a$  的通信范围内有  $I$  个其他传感器节点,令  $\Sigma = \{\phi_i | 1 \leq i \leq I\}$

代表这  $I$  个传感器节点的集合。由于这  $I$  个节点距离节点  $a$  的远近不同,为合理利用节点能量,提高节点生存期,考虑这  $I$  个节点与  $a$  的当前距离对节点转发优先级的影响,得到节点数据转发优先级  $T_{ia}$ ,有:

$$T_{ia} = \omega(D_{ia}) P_i \quad (11)$$

式中,  $\omega(D_{ia})$  为当前距离影响函数。通过全球定位系统 GPS, 节点  $a$  可获知这  $I$  个节点目前位置坐标。设节点  $a$  当前位置为  $(x_1, y_1)$ , 节点  $i$  当前位置为  $(x_2, y_2)$ , 根据平面两点距离公式,可知两节点间距离  $l_{ia}$  为:

$$l_{ia} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (12)$$

设节点通信阈值为  $d_0$ , 最大通信半径为  $d$ , 将  $l_{ia}$  引入式(8), 可得:

$$\omega(D_{ia})' = \begin{cases} 1 - \sqrt{\frac{l_{ia}}{d}}, & l \in (0, d_0] \\ (1 - \sqrt{\frac{l_{ia}}{d}}) \frac{(l_{ia} - d)^4}{(d_0 - d)^4}, & l \in (d_0, d] \end{cases} \quad (13)$$

考虑到  $\omega(D_{ia})$  不是影响数据消息转发的唯一因素, 这里将比例在计算中的权重适当缩小, 因此可以得到当前距离影响因子  $\bar{\omega}(D_{ia})$ :

$$\bar{\omega}(D_{ia}) = e^{\omega(D_{ia})'} / e \quad (14)$$

在数据转发过程中, 节点  $a$  把数据直接发送给  $I$  中所有满足  $T_{ia} > P_a$  的节点。

## 2.5 队列管理机制

显然, 上述数据转发策略算法会导致一个消息产生多个复本, 因此本文引入队列管理机制管理这些复本。本文构造与节点传输优先级  $P_i$  相关的动态消息队列, 并引入消息生存时间决定消息传递的优先顺序及消息丢弃原则。

每个传感器节点均有一个消息队列保存准备要发送的数据消息, 这些消息有 3 个来源<sup>[9]</sup>: (1) 节点收集的数据; (2) 节点收到其他节点转发来的消息; (3) 节点把消息转发到非汇聚点节点后, 保存转发出去的消息中属于自身产生的那部分消息的复本。

由于不同节点与汇聚点发生数据传输的概率不同, 即传输优先级  $P_i$  不同, 显然  $P_i$  越高的节点有更高的概率把消息发送到汇聚点, 因此  $P_i$  越高的节点有更大容量的消息队列,  $P_i$  越低的节点消息队列容量更小。令网络初始时, 每一节点的消息队列能容纳相同数目的消息  $N$ , 网络中允许节点最小消息队列容量为  $\gamma$ 。在网络运行任一时刻, 对任一传输优先级为  $P_i$  的节点  $i$ , 队列容量为  $N_i$  有:

$$N_i = \lceil 2(N - \gamma)P_i \rceil + \gamma \quad (15)$$

消息生存时间 ST 即消息在网络中的存活时间。消息的 ST 越长, 越有可能在网络中存在更多的复本, 已被汇聚节点接收的可能性也越大。因此运用生存时间表示网络中各个消息的冗余程度和重要性<sup>[11]</sup>。每个消息的头部均包含一个域, 存放消息的生存时间值。当消息初始化时, ST 被初始化为 0。每个传感器节点中均有一个本地计时器, 每当计时器超时, 节点消息队列中所有消息的生存时间值将被增加。

虽然各节点队列容量不一, 但都按 ST 由短到长的顺序进行排列, ST 短的消息排在前面并得到优先发送权。以下两种情况会发生消息丢弃现象: (1) 当消息到达传感器节点时存储队列已满时, 若新消息的 ST 值大于队尾消息 ST 值, 则丢弃新消息, 否则丢弃队尾消息; (2) 队列若某消息 ST 值大于

整个网络延迟容忍限度, 则丢弃此消息。

## 3 仿真实验

本文对 NPD, SRAD, DT 和 Flooding 进行仿真实验, 从以下 3 方面进行性能比较。

1) 比较 NPD 与其他 3 种算法针对数据消息的平均传输成功率、平均传输能耗和平均传输延迟;

2) 研究不同的实验参数对这 4 种算法产生的影响;

3) 分析比较 NPD 与其他 3 种算法的网络寿命。

仿真实验中, 定义所有传感器节点在一次运动过程中运动速度保持恒定。假设节点的数据产生过程遵循平均到达时间间隔为 50s 的泊松过程, 其他网络参数以及相应的缺省值见表 1。需强调的是, 对于 Flooding 算法, 当节点的存储队列满后, 节点不再接收任何邻居节点发送来的消息。以下的实验结果如未特别说明, 均为 100 次独立实验结果的均值。

表 1 实验环境参数

参数	缺省值
网络尺寸	200×200
传感器节点数量	100
通信半径 d(m)	3
通信阈值 $d_0$ (m)	2
节点速度(m/s)	1~5
节点队列长度初始值	200 个消息
节点队列最小长度	100 个消息
消息大小(bit)	200
汇聚点位置	(100, 100)
NPD 的 $\alpha, \beta, \gamma$	0.7, 0.2, 0.1
消息生成速度	0.02/s
节点初始能量(J)	10
GPS 通信消耗能量(J/Per)	0.25
GPS 通信时间间隔(s)	2.5
最长延迟容忍时间(s)	3200

### 3.1 4 种算法性能对比

将 DT, Flooding, SRAD 及 NPD 分别在默认实验参数下进行仿真, 就数据交付率、平均副本数以及平均延迟 3 个性能指标进行比较。实验结果如表 2 所列。

表 2 默认参数下仿真结果的对比

	DT	Flooding	SRAD	NPD
数据交付率(%)	58.3	66.4	91.2	91.3
消息平均副本数	1	9.64	6.59	6.11
平均延迟(s)	840.3	728.6	136.4	112.6

通过表 2 可以看出, NPD 和 SRAD 有着最高的数据交付率。对于 DT, 由于传感器节点不与汇聚点相遇, 节点的消息永远不会被汇聚点接收, 因此交付率最低, 而 Flooding 的消息由于能够被不同节点传输到汇聚点, 因此具有比 DT 更高的交付成功率。NPD 充分考虑到节点的优先级来进行数据转发, 并有着动态的队列容量, 因此消息平均副本数较 SRAD 更低, 同时, NPD 考虑到节点的速度对节点优先级的影响, 因此有着最低的平均延迟。

### 3.2 节点密度对性能的影响

网络的拓扑结构与节点的密度相关, 本组实验主要研究不同节点密度情况下不同算法的性能。节点密度的变化通过改变节点的总数来实现, 实验结果如图 2(a)~(c) 所示。

从图 2(a) 可以看出, NPD 和 SRAD 均有较高数据传输成功率, 同时 4 种算法传输成功率与节点密度无关。图 2(b) 显

表3 默认参数下各算法的网络寿命

	DT	Flooding	SRAD	NPD
网络寿命(天)	581	0.52	1.92	2.56

示随着节点密度的增大,满足传输条件的邻居节点数量增加,NPD,SRAD,Flooding的平均消息副本数均有所上升。NPD构建了与节点传输优先级相关的动态消息队列,因此一直保持较低的消息副本数;图2(c)讨论了节点的密度与平均传输延迟的关系。节点密度增大有利于减少消息从数据源节点到汇聚点的跳数,因此降低了NPD和SRAD的传输延迟,进一步NPD在传输优先级计算中充分考虑了节点的运动速度,使得NPD有着相对更低的传输延迟。

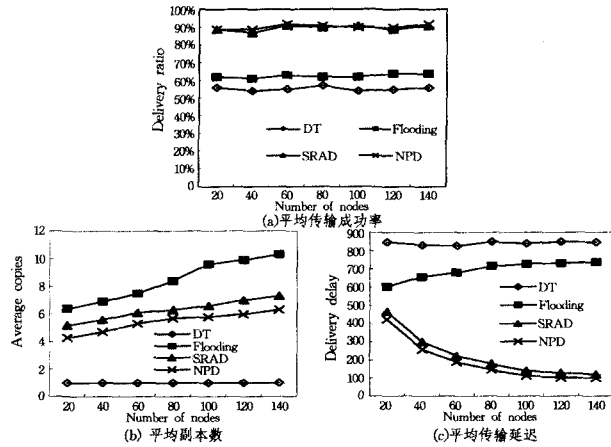


图2 节点密度的影响

### 3.3 通信半径对性能的影响

图3表现的是节点最大通信半径从1逐渐增大到8时4种算法的性能对比。可看出,当通信半径增加时,各算法的消息传输成功率均不断增加,而NPD与SRAD的传输成功率远高于DT和Flooding,尤其当节点的通信半径较小时更为明显。图3(b)表明,因为传感器节点只与汇聚点发生直接通信,直接传递算法的平均消息副本数永远为1。随着通信半径的增大,NPD平均消息副本数远小于SRAD和Flooding。图3(c)显示,随着通信半径的增大,各算法平均传输延迟均逐渐减小,因为通信半径增大会增大消息传输到汇聚点的概率。由于NPD有效地选择了与更高的数据转发优先级的节点进行通信,因此它的平均传输延迟最低。

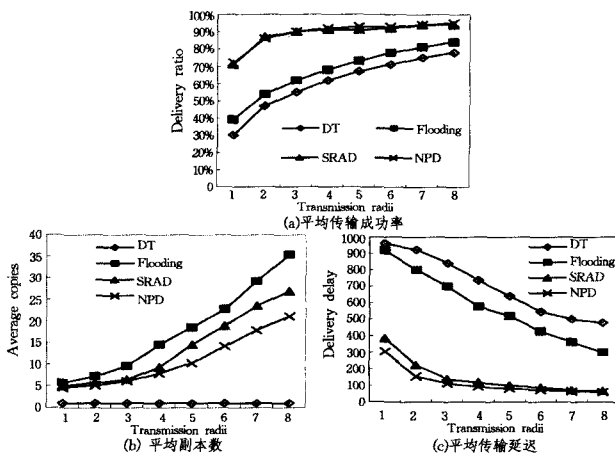


图3 通信半径的影响

### 3.4 网络寿命

本组实验研究4种算法的网络寿命,并从总体能耗方面对算法进行分析。实验结果如表3所列。

从表3可看到,4种算法中DT的网络寿命最长,这是因为DT中节点不与其他节点发生消息通信,只在遇到汇聚点时将队列中的消息递交,所以节点的能量消耗极小。而对于其他3种算法,由于NPD在计算节点优先级时充分考虑到节点当前剩余能量,并且NPD能有效地控制消息平均副本数,使节点能量消耗降低,因此NPD策略有着较高的网络寿命。

**结束语** 为了更有效地收集DTMSN中的数据,本文提出了一种基于节点优先级的动态数据传输策略NPD。与已有工作相比,NPD的主要贡献表现在以下两方面:(1)提出了一种有效的动态路由机制,通过节点间距离、节点剩余能量、节点运动速率得到节点传输优先级,并根据节点优先级进行数据转发;(2)NPD引入有效的动态消息队列管理机制,有效地控制了消息副本数量,降低了网络能耗,延长了网络寿命。

仿真实验验证了NPD的优异性能,与现有几种算法相比,NPD的数据传输成功率高,传输能耗低且延迟较小。

### 参考文献

- [1] 俞靓,王志波,骆吉安,等. 面向移动目标追踪的无线传感器网络QoS指标体系设计[J]. 计算机学报, 2009, 32(3): 441-462
- [2] Burleigh S, Hooke A, Torgerson L, et al. Delay-Tolerant networking-an approach to interplanetary Internet[J]. IEEE Communications Magazine, 2003, 41(6): 128-136
- [3] Fall K. A delay-tolerant network architecture for challenged internets [C]// Feldmann A, eds. Proc. of the ACM SIGCOMM 2003 Conf. on Computer Communications. New York: ACM Press, 2003: 27-34
- [4] Leguay J, Friedman T, Conan V. DTN routing in a mobility pattern space[C]// Guerin R, ed. Proc. of the ACM Workshop on Delay Tolerant Networking and Related Topics (SIGCOMM 2005). New York: ACM Press, 2005: 276-283
- [5] Wang Yu, Wu Hong-yi. The Delay/Fault-Tolerant mobile sensor network(DFT-MSN): A new paradigm for pervasive information gathering[J]. IEEE Trans. on Mobile Computing, 2007, 6(9): 1021-1034
- [6] Vahdat A, Becker D. Epidemic routing for partially connected ad hoc networks[R]. CS-200006. 2000
- [7] Wang Yu, Wu Hong-yi. Replication-Based efficient data delivery scheme (RED) for delay/fault-tolerant mobile sensor network (DFT-MSN) [C]// Gregori E, ed. Proc. of the 4th Annual IEEE Int'l Conf. on Pervasive Computing and Communications Workshops. Washington: IEEE Computer Society Press, 2006: 485-489
- [8] 朱金奇,刘明,龚海刚,等. 延迟容忍移动传感器网络中基于选择复制的数据传输[J]. 软件学报, 2009, 20(8): 2227-2240
- [9] 张可,曾家智,刘伟. 延迟容忍移动传感器网络中基于概率复制的数据传输策略及其性能研究[J]. 电子与信息学报, 2010, 32(3): 677-681
- [10] Heinzelman W R. An application-specific protocol architecture for wireless microsensor networks[J]. IEEE Trans. on Wireless Communications, 2002, 1(4): 660-670
- [11] 许富龙,刘明,龚海刚,等. 延迟容忍传感器网络基于相对距离的数据传输[J]. 软件学报, 2010, 21(3): 490-504