一种改进二进制防碰撞算法研究 *

周艳聪[1,2], 孙晓晨[3], 顾军华[2]
(1. 天津商业大学 信息工程学院, 天津 300134; 2. 河北工业大学 电气工程学院, 天津 300130; 3. 天津大学 理学院, 天津 300072)

摘　要: 为了克服普通二进制算法交互次数多、通信数据量大的缺点,提出了一种改进二进制防碰撞算法。新算法首先进行信息预处理,识别过程中只处理冲突位,并以待识别标签为叶子节点反向构建识别树,在叶子节点的父层实现标签识别。其在碰撞位连续或间断情况下均可有效认读标签。尤其当冲突标签数量较多而碰撞位明显小于标签长度时,其在交互次数、通信数据量和识别效率上均表现出较明显的优势。模拟仿真、理论分析和算法实例验证了新算法的可行性、实用性和高效性。

关键词: 防碰撞算法; 二进制搜索; 射频识别; 预处理; 反向搜索
中图分类号: TP301.6　　**文献标志码**: A　　**文章编号**: 1001-3695(2012)01-0256-04
doi:10.3969/j.issn.1001-3695.2012.01.071

Research on improved binary anti-collision algorithm

ZHOU Yan-cong[1,2], SUN Xiao-chen[3], GU Jun-hua[2]
(1. *School of Information Engineering*, *Tianjin University of Commerce*, *Tianjin* 300134, *China*; 2. *School of Electrical Engineering*, *Hebei University of Technology*, *Tianjin* 300130, *China*; 3. *College of Science*, *Tianjin University*, *Tianjin* 300072, *China*)

Abstract: To conquer the shortcomings of general binary anti-collision algorithms, such as more interactive times and communication data, proposed an improved binary anti-collision algorithm. When tags were identified the IDs information firstly and then in the whole process only the collision bits were dealt with. It constructed the identification tree from tags leaves and could identified the tags directly on leaves' father level. No matter whether the tags' collision bits were consecutive or not, the algorithm was always effective. On interactive times, communication data and identification efficiency, the new algorithm had obvious superiority, especially when there were many tags within the scope of reader and collision bits were less than the length of tags. The algorithm is proved feasible, practical and effective by simulation, theory analysis and testing instance.

Key words: anti-collision algorithm; binary-tree searching; radio frequency identification (RFID); pretreatment; converse searching

0　引言

射频识别(radio frequency identification, RFID)是 20 世纪 90 年代兴起的一种自动识别技术。目前已应用在收费站、停车场、超市、地下管线、企业考勤等诸多方面,并且成为未来物联网的主要技术之一。RFID 系统一般由阅读器和电子标签两部分组成。根据工作频段的不同,系统可分为低频、高频和超高频段三种。其工作原理是射频信号通过空间耦合即交变磁场或电磁场实现无接触信息传递,从而达到通信和物体识别的目的。当多个标签同时处于阅读器范围内,并向阅读器传递信号时,就发生了识别冲突,即碰撞。碰撞是射频识别系统中的一个常见问题,如何有效解决该问题成为系统的关键。

对于标签识别过程中的碰撞问题,目前的解决方案主要有四种:时分多址、码分多址、频分多址和空分多址,其中时分多址最为常见。基于时分多址的标签防碰撞算法主要有以 ALOHA 算法为代表的随机防碰撞算法和以二进制防碰撞算法为代表的确定型防碰撞算法。

1　基于二进制的防碰撞方案

对于二进制防碰撞算法,国内外很多专家学者进行了大量研究,主要是在 ISO/IEC 18000—6B 的二进制防碰撞算法的基础上进行改进。二进制防碰撞算法的基本思想是每次将碰撞标签按照最高碰撞位上的值的不同分成两个子集 0 和 1,先查询最高碰撞位为 0 的子集,若不发生碰撞,则正确识别标签;若仍有碰撞,则根据该次碰撞的首个碰撞位再分裂,进一步把子集 0 分成 00 和 01 两个子集;依次类推,直到识别出子集 0 中的所有标签,按同样的步骤查询最高碰撞位为 1 的子集,直到所有标签被正确识别。

对于基于二进制的防碰撞算法,在国外的研究中典型的有 BBT(bit-by-bit tree)算法[1]、Q-Tree 算法[2]、ABS(adaptive binary splitting)算法[3]等。国内的研究主要有:a)基于后退式的二进制树搜索算法[4],当识别出一个标签后,算法根据上一次请求命令参数来获得下一个请求命令,以此来极大地缩短识别过程;b)动态二进制算法[5],主要针对基本算法中阅读器命令和

收稿日期: 2011-05-31; **修回日期**: 2011-07-21　　**基金项目**: 国家自然科学基金资助项目(60872161,70871090);天津市应用基础及前沿技术研究计划项目(11JCYBJC00200);科技人员服务企业行动资助项目(2009GJA20014)

作者简介:周艳聪(1978-),女,河北饶阳人,讲师,博士研究生,主要研究方向为物流优化、数据挖掘(zycong78@126.com);孙晓晨(1978-),男,副教授,博士,主要研究方向为随机系统建模与优化、算法分析与设计;顾军华(1966-),男,教授,博士,主要研究方向为数据挖掘、图像处理.

标签响应信息中存在大量冗余数据的情况而提出,改进算法只需要阅读器传送碰撞位之前的 K 位,标签传送命令外的 $L-K$ 位数据,从而大大降低了通信量,使通信效率提高了 50%;c)多状态的二进制防碰撞算法[6],其引入了休眠计数的方法将待识别标签分别进行处理,缩小了标签识别范围,减少了识别过程中的数据传送量;d)对标签设计的改进和相应的分组识别算法[7],利用各标签与阅读器距离不同这一物理条件对发生冲突的标签进行分组,把数量大的冲突标签分为多个小部分,从而提高识别效率;e)针对终端射频卡数量越多信息位数越多、传送时间越长的缺点提出的二进制防碰撞算法的改进算法[8],其使得在标签位数逐渐增多的情况下,查询次数增加速度明显减慢;f)针对标签号连续和碰撞连续情况下标签识别效率较低的缺点提出的轮询算法[9,10],大大提高了该情况下的标签识别效率。

对于目前较常见的几种防碰撞方案的基本思想和识别步骤简要描述如下:

a)基本二进制防碰撞算法。其主要以序列号(UID)识别标签为基础。首先阅读器发送一个全“1”的请求命令,所有标签收到该命令后都将回传自身序列号,阅读器通过收到的回应判断是否存在碰撞。若存在碰撞,则根据识别出的碰撞位调整发送命令,直到正确识别所有标签。期间标签序列号小于等于阅读器命令的标签作出响应。在该过程中起决定作用的是阅读器所使用的信号编码,其必须能够确定碰撞的准确位置,本文讨论方案均采用曼彻斯特编码。基本二进制算法识别标签的步骤如下:

(a)当标签进入阅读器的工作范围时,阅读器发出一个最大序列号,全“1”指令,所有标签响应并传输自身序列号到阅读器。

(b)阅读器检测接收到的标签序列号的每一位,出现不一致时,则判断该位有碰撞。

(c)确定发生碰撞后,将命令中对应最高冲突位置 0,将其作为新的命令序列号发送给各标签,标签按照约定回应,阅读器判断有无碰撞。若存在碰撞时,阅读器继续修改命令序列号并发送;若不存在碰撞时,阅读器可直接确定出要选择的标签序列号。

(d)阅读器选择某序列号后,直接识读标签信息,之后使该标签进入“去选择”状态,使得标签即使在阅读器范围内也不再响应,除非有阅读器重新进入激活标签。

(e)重复流程(a)~(d),识别出下一个序列号。

(f)多次识别后可完成所有标签的读取。

将基本二进制识别标签的过程用二叉树来描述会发现,其每识别一个标签,即达到某个叶子节点后,将重新从根节点开始搜索下一个待识别标签,直到所有叶子节点全部识别。

b)后退二进制防碰撞算法。从基本二进制的识别过程不难发现,每次从根节点重新搜索将大大增加标签的识别路径,从而降低识别效率。后退二进制防碰撞算法就是针对该缺点在基本二进制算法的基础上进行的一种改进。主要改进点为:后退二进制算法完成一个标签的识别后不返回根节点,而是从离此标签最近的有待识别标签的内部节点开始搜索,直到该分支点以下标签全部搜索完毕,然后继续向上返回进行搜索。具体识别过程如表 1 所示。

c)动态二进制防碰撞算法。由基本二进制防碰撞算法的识别流程可知,阅读器发送的命令自最高碰撞位后为全“1”,这些全“1”位数据对标签而言没有任何意义。同样,当标签回传序列号时,序列号总是被完整地传输,而最高碰撞之前的所有数据位对阅读器而言没有任何意义。当标签序列号较长时,其冗余传输数据量是相当可观的,势必影响整体识别效率。因此,阅读器发送的命令序列号中最高碰撞位后的数据位和标签回传信息中最高碰撞位之前的数据位完全可以省略,均不必完整传输。动态二进制就是针对该点在基本二进制防碰撞算法基础上做的改进。其具体操作流程为:阅读器在 Request 命令中只发送要搜索序列号的已知部分($L\sim X$)作为搜索的依据,X 为当前最高碰撞位。所有($L\sim X$)位序列号与 Request 命令中序列号前缀相符的标签,回传序列号的剩余($X-1$)~0 位作为应答。由于命令序列号和回应的标签序列号是动态改变的,故这种算法称为动态二进制防碰撞算法。第一次问询时,阅读器发送 Request(NUL,L),处于阅读器作用范围内的所有标签都回应,以便探测冲突位,生成后续的命令序列。详细识别过程如表 2 所示。

表 1　后退二进制算法识别过程

	Request (11111111)	Request (01101101)	Request (01001101)	Request (01001001)	Request (01001101)	…	Request (11111111)
响应	X1X01X01	01X01X01	01001X01	Tag 1 识别	Tag 2 识别	…	Tag 8 识别
Tag1	01001001	01001001	01001001	01001001	无声状态	…	无声状态
Tag2	01001101	01001101	01001101	——	01001101	…	无声状态
Tag 3	01101001	01101001	——	——	——	…	无声状态
Tag 4	01101101	01101101	——	——	——	…	无声状态
Tag 5	11001001	——	——	——	——	…	无声状态
Tag 6	11001101	——	——	——	——	…	无声状态
Tag 7	11101001	——	——	——	——	…	无声状态
Tag 8	11101101	——	——	——	——	…	11101101

表 2　动态二进制算法识别过程

	Request (NUL,8)	Request (0,7)	Request (010,5)	Request (010010,2)	…	Request (NUL,8)
响应	X1X01X01	1X01X01	01X01	Tag 1 识别	…	Tag 8 识别
Tag1	01001001	1001001	01001	01	…	无声状态
Tag2	01001101	1001101	01101	——	…	无声状态
Tag 3	01101001	1101001	——	——	…	无声状态
Tag 4	01101101	1101101	——	——	…	无声状态
Tag 5	11001001	——	——	——	…	无声状态
Tag 6	11001101	——	——	——	…	无声状态
Tag 7	11101001	——	——	——	…	无声状态
Tag 8	11101101	——	——	——	…	11101101

本文将主要对二进制防碰撞算法进行研究,并针对如何减少识别过程的交互次数和降低识别冲突过程通信量进行改进,在基本二进制防碰撞算法的基础上,结合动态二进制防碰撞算法的特点,提出一种改进的二进制防碰撞方案。

2　改进的二进制防碰撞方案

2.1　算法基本思想

目前的几种二进制防碰撞算法的基本思路是按照某种规则自顶向下构建识别树,其中问询过程对应中间节点的构建和访问,待识别标签对应识别树的叶子节点。算法目标就是在访问所有叶子的前提下尽可能减少交互次数,即识别树中间节点个数和访问次数尽可能少,同时减少数据传输量。新的改进二进制防碰撞算法从冲突位入手,标签每次只传递冲突末位数据,并以标签冲突位数据为叶子节点,自底向上构建识别树,在

识别树的最低两层即可实现叶子的识别。该方法大大减少了中间节点的构建,有效降低了识别树的高度,缩短了识别过程,减少了访问次数;同时,除去个别指令,标签只向阅读器传递最低冲突位数据,这样大大减少了通信数据量,从而有效提高了算法识别效率。

2.2　算法约定

与基本二进制防碰撞算法类似,本算法需要四个基本命令(Request(CUID)、Select(UID)、RW-Data 及 Unselect)和一个冲突位设定命令(CollisionBitsSet(CID))。

a)Request(CUID)请求命令。CUID 为二进制数,位数为冲突位数减 1,初始值为全“0”。阅读器首先发送 Request(CUID)命令;区域内标签先对接收命令末尾连接“1”,然后根据冲突模板处理自身序列号,小于等于处理后的命令序列号的标签回送序列号最低冲突位,反之,则不回应。阅读器每发送一次 Request(CUID),CUID 累加 1,作为下一次的命令序列号。

b)Select(UID)选择命令。用某个事先确定的 UID 作为参数发送给标签。具有相同 UID 的标签被选择,以便进行后继的读出和写入数据命令。其他序列号标签只对 Request 命令应答。

c)RW-Data 读写命令。阅读器完成对选中标签的数据的读写操作。

d)Unselect 去选择命令。取消被选中的标签,使其进入“去选择”状态。该状态下标签对 Request 命令不回应。要重新激活标签,必须重新进入阅读器的作用范围内,以实行复位。

e)CollisionBitsSet(CID)冲突位设定命令。由阅读器将提取到的冲突位信息发至各标签,各标签据此设定数据发送模板,保证标签比较数据仅为冲突位信息;信息返回时只返回最低冲突位即可。

该算法的实现必须满足以下三个条件:a)选用适当的基带编码,以便于识别碰撞,本文采用曼彻斯特编码,其碰撞位识别原理如图 1 所示;b)每个标签拥有唯一的序列号;c)阅读器作用范围内标签能在同一时刻传送其序列号,以便准确监测碰撞位的发生。除此外还有如下约定:因为二进制位上的取值具有相互排斥特性(非 1 即 0),故当阅读器探测到只有 1 位发生碰撞时,将不再发送请求命令而自动识别出这两个标签。

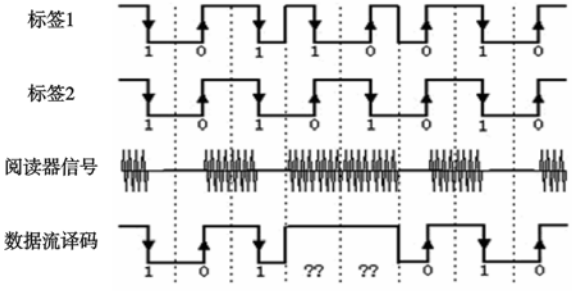

图1　曼彻斯特编码碰撞位识别原理

待识别标签数量较大时,此算法在标签碰撞位连续或不连续情况下都可以大大减少交互次数,提高识别速度;但当碰撞位较多而标签数量较少时,效果不是很明显。

2.3　算法流程

该算法识别标签的大致过程可描述如下:

a)阅读器发送 Request(1)命令,令识别范围内所有标签回应。根据回应结果一次检测出所有冲突位,依此设定冲突模板 CID。CID 为二进制数据,与标签序列号位数相同,对应冲突位均置 0,其他位置 1。

b)阅读器初始化命令序列号(CUID)。根据冲突模板标注的冲突位,除去最低冲突位,剩余冲突位按照原来顺序组成命令序列号 CUID。

c)阅读器向各标签发送 CollisionBitsSet(CID),标签根据 CID 提取比较数据位,确保接收阅读器命令后进行数据比较时只比较冲突位信息。

d)阅读器发送 Request(CUID),CUID 自动累加 1,标签接收并还原命令,还原命令 = 命令序列号 &“1”,并根据提取的比较数据位处理自己的序列号,小于等于还原命令序列时,发送标签最低冲突位;否则,不回应。

e)阅读器接收到的信息要么没有冲突,要么只有一位冲突,这时可直接识别。

f)当 CUID 重新为全“0”时,识别结束;否则,转 d)继续识别过程。

详细算法流程如图 2 所示。

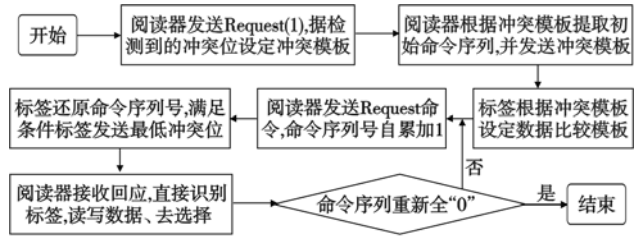

图2　改进二进制防碰撞算法流程

2.4　算法示例

下面以长度为 8 且碰撞位不连续的下述 8 个标签为例说明算法识别过程。标签序列号分别为 01001001、01001101、01101001、01101101、11001001、11001101、11101001、11101101。为了对比识别过程,将分别应用后退二进制、动态二进制和改进二进制对该 8 个标签进行识别。后退二进制防碰撞算法、动态二进制防碰撞算法和改进二进制防碰撞算法对 8 个标签的识别过程分别如表 1 ~3 所示。其中,后退二进制算法识别该 8 个标签需要发送 15 次查询命令,动态二进制算法需要发送 20 次查询命令;如果采用后退策略,则只需要发送 15 次查询命令。

改进二进制算法对 8 个标签的识别过程:阅读器发送 Request(1),所有标签响应,阅读器判断出该次探询的冲突位为 2,5,7 位;冲突模板 CID 为“01011011”,除去最低冲突位,剩余两位组成命令序列 CUID,初始为“00”。标签接收到两位的命令序列后,末尾连接“1”,将其还原为 3 位命令序列,得到“001”,最低冲突位永远置 1;然后根据冲突模板处理自身序列号,取出对应冲突位(2,5,7 位)与还原后的命令序列比较,小于等于时回传最低冲突位,否则不回应;每发送一次 Request(CUID)命令,CUID 累加 1,即第二次命令序列自动变为“01”,重复上述处理过程,当 CUID 累加到“00”时,处理结束。详细识别过程如表 3 所示。

改进二进制算法识别该 8 个标签需要发送 5 次查询命令,信息通信量仅为 81。当冲突标签数量较多、冲突位相对较少时,这种优势尤其显著。

3　实验仿真与分析

与基本二进制和后退二进制、动态后退二进制防碰撞算法相比,该防碰撞算法在标签数量较大时,碰撞连续发生或不连续发生均可以极大提高标签的识别速度。在 MATLAB 中对算

法进行模拟仿真,仿真参数如表 4 所示。在识别标签位数为 16 位,其中第 3、4、7、9、13 上的数据各标签相同,识别标签数量分别为 8、64、128、256、512、1024、2048 时,其交互次数对比如图 3 所示,数据通信量(单位:KB)对比如表 5 所示,其中数据均为多次运行程序的平均值。

表 3　改进二进制算法识别过程

	Request(1)	Request(00)	Request(01)	Request(10)	Request(11)
响应	X1X01X01	X Tag 1,2 识别	X Tag 3,4 识别	X Tag 5,6 识别	X Tag 7,8 识别
Tag1	01001001	0	无声状态	——	无声状态
Tag2	01001101	1	无声状态	——	无声状态
Tag 3	01101001	——	0	无声状态	无声状态
Tag 4	01101101	——	1	无声状态	无声状态
Tag 5	11001001	——	——	0	无声状态
Tag 6	11001101	——	——	1	无声状态
Tag 7	11101001	——	——	——	0
Tag 8	11101101	——	——	——	1

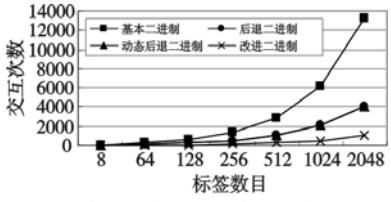

图3　各算法标签识别交互次数对比

表 4　仿真参数

运行环境		标签信息		
OS	WinXP	长度	冲突位	标签数量
CPU	1.85 GHz	16 位	0,1,2,5,6,8,10,11,	8,64,128,256,512
内存	512 MB		12,14,15	1024,2048

表 5　各算法标签识别通信数据量对比　/KB

	8	64	128	256	512	1024	2048
基本二进制	0.154	6.882	26.242	102.016	401.408	1590.784	6330.368
后退二进制	0.096	1.152	2.506	5.63	12.286	26.622	57.342
动态后退二进制	0.062	0.714	1.609	3.365	7.26	16.316	36.476
改进二进制	0.018	0.156	0.32	0.654	1.342	2.75	5.63

从图中不难看出,改进二进制算法较其他二进制算法在通信效率上有较大的优势。

对几种方法进行算法分析和比较,以算例中 8 个标签的识别为例,阅读器对标签的问询次数和通信数据量(单位:bit)对比如表 6 所示。当冲突标签数量较多时,其优势对比更加明显;当碰撞位较多而碰撞标签数量较少时,其优势不明显,而且冲突模板的加入也增加了标签设计的复杂度。

从图 3、表 5 和表 6 可以看出,改进算法无论在识别效率、交互次数和通信数据量上都有较大的优势。假设标签长度为 L,待识别标签数目为 N,存在 M 个碰撞位,考虑最复杂情况下,即碰撞标签数目最大的情况($N=2^M$)。

对于交互次数,基本二进制算法识别全部标签需要的交互次数为 $N+\frac{MN}{2}=2^{M-1}(2+M)$;后退二进制算法识别全部标签需要的交互次数为 $2N-1=2^{M+1}-1$;改进二进制算法识别全部标签需要的交互次数只与冲突位数有关,其交互次数为 $2^{M-1}+1$。即使每位都存在冲突,即 $L=M$,改进二进制的交互次数也仅为后退二进制算法交互次数的 1/4,计算公式为 $\frac{2^{M-1}+1}{2^{M+1}-1}\approx\frac{1}{4}$;为传统二进制算法交互次数的 $1/(2+M)$,计算公式为 $\frac{2^{M-1}+1}{2^{M-1}(2+M)}\approx\frac{1}{2+M}$。

表 6　算例交互次数和通信数据量对比

算法名称	问询次数	通信数据量/bit
基本二进制	20	608
后退二进制	15	376
动态后退二进制	15	286
改进二进制	5	81

对于通信数据量,基本二进制算法的通信量为 $L(\frac{3}{4}N^2+\frac{3}{4}MN+\frac{5}{4}N)=L(\frac{3}{4}2^{2M}+\frac{3}{4}M2^M+\frac{5}{4}2^M)$;后退二进制算法的通信量为 $L(N+MN+2N-1)=L[(3+M)2^M-1]$;改进二进制算法的通信量为 $1+LN+(M+1)2^{M-1}=1+2^ML+(M+1)2^{M-1}$。

同样,为了方便讨论,考虑每位都存在冲突的情况,即 $L=M$。改进二进制算法的通信量与基本二进制算法的通信量之比约为 $\frac{6M+2}{3M^2+3M\times2^M+5M}$;与后退二进制算法的通信量之比约为 $\frac{(3M+1)2^{M-1}}{M[(3+M)2^M-1]}$;动态后退二进制算法的通信量约为后退二进制算法的通信量的 1/2,故改进二进制算法的通信数据量与动态后退二进制算法的通信量之比约为 $\frac{(3M+1)2^M}{M[(3+M)2^M-1]}$。

当碰撞位数较少($M<L/2$)而待识别标签较多时,即 $N=2^M$,改进二进制算法的通信量较其他三种算法会有更明显的优势。可见,该算法是一种有效的标签防碰撞算法。

4　结束语

通过对二进制防碰撞算法的研究分析,结合后退二进制和动态二进制算法的优点,提出了一种改进的二进制防碰撞算法。该方法引入信息预处理机制,识别过程中只处理冲突位信息,以待识别标签冲突位为叶子节点建立识别树,在叶子节点的父层直接实现叶子节点的识别,这在一定程度上减少了中间无效节点的产生和访问,从而大大提高了标签识别效率。模拟仿真、理论分析以及大量的实验数据表明,改进二进制防碰撞算法能够将识别区域内的多个标签准确地识别出来,与其他二进制防碰撞算法相比,该算法无论在通信数据量、识别效率和交互次数上都有较大的优势。当待识别标签数量较大,$N=2^M$ 时,其交互次数较后退二进制算法可以提高将近 3 倍。当冲突位数明显少于标签位数时,在数据通信量上改进二进制算法凸显出其绝对优势。

参考文献:

[1] HUSH D R, WOOD C. Analysis of tree algorithms for RFID arbitration[C]//Proc of IEEE International Symposium on Information Theory. 1998:107.

[2] LAW C, LEE K, SIU K Y. Efficient memoryless protocol for tag identification[C]//Proc of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications. New York: ACM Press, 2000:75-84.

(下转第 262 页)

能量存储于能量消耗维稳期,计算出的理想状态下簇间最佳通信距离 d_{best},距簇首距离小于 d_{best},若满足 $d_{i-sink}^2 < d_{i-j}^2 + d_{j-sink}^2$ 的下一跳节点数大于 1 时,选择 $|d_{i-j} - d_{best}|$ 最小的节点作为下一跳节点。

3　仿真结果

3.1　实验参数

本文在 MATLAB 仿真平台进行实验。所有节点初始状态相同,150 个节点随机分布在 200 m×200 m 的区域内,数据传输轮次 800 轮。仿真参数如表 1 所示。

表 1　仿真参数

主要参数	参数值	主要参数	参数值
节点初始能量 E_0	0.5	E_{mp}	0.0013 pJ/bit/m4
基站位置	(100,100)	E_{fs}	10 pJ/bit/m^2
数据包大小	4 000 bit	C	0.001
E_{elect}	50 nJ/bit		

图 4 为仿真过程中全网节点总能量补给模型,将时间换算为轮次。0~300 轮,能量补给值随时间上升;300~500 轮,能量补给值呈基本稳定状态;500~800 轮,能量补给值随时间下降。

3.2　实验结果分析

图 5 为本文 ACSP 算法与 LEACH、PHC 算法在同样的梯形能量补给模型下,全网节点剩余能量与轮次曲线图。可看出,耗能期,ACSP 更加节能;到 200 轮左右,补给能量刚好可维持全网节点工作而无须消耗电池能量;随着采能率升高,补给能量出现剩余,由于全网节点平均剩余能量未达到设定的阈值水平,所以节点仍以节能模式工作,多余的补给能量用于电池储能,为下一个耗能期做准备;当达到设定阈值时,进入稳定期,可维持储能与耗能稳定,避免能量溢出浪费;620 轮左右,再次进入耗能期。

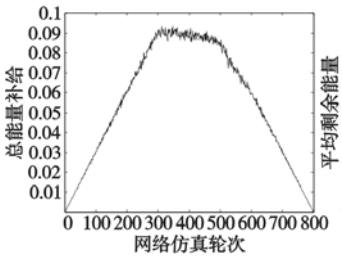

图4　全网节点总能量补给模型

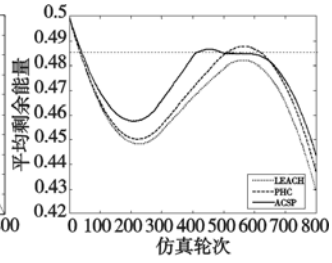

图5　节点每轮平均剩余能量

图 6~8 分别为 ACSP、LEACH、PHC 算法在同样能量补给模型下全网节点剩余能量图。可看出 800 轮数据传输后,图 7、8 节点剩余能量值范围较广,ACSP 算法节点剩余能量值较 LEACH、PHC 算法更集中,说明 ACSP 算法较 LEACH、PHC 算法能耗更均衡。

4　结束语

本文提出一种自适应太阳能补给的 WSN 分簇路由算法 ACSP,做出了三点创新:a)分析了太阳光照的特点,设计了太阳能能量补给模型,并根据能量补给特点将网络分为三个时期:耗能期、储能期、稳定期;b)能量补给充足时,计算最佳通信距离作为路由选择的基准;c)根据最佳通信距离估算出簇首节点占总节点数百分比,并将节点能量补给速率、能耗速率、剩余能量值三个因素作为簇首选择权衡因子,选出最佳簇首。理论和实验表明,ACSP 算法能实时适应网络中能量补给情况,充分利用补给能量,避免了能量溢出浪费,且节点能耗较 LEACH、PHC 算法更均衡。

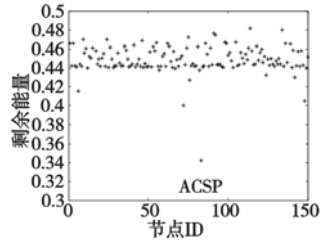

图6　ACSP算法节点剩余能量

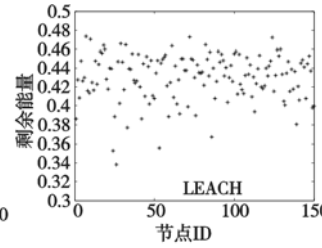

图7　LEACH算法节点剩余能量

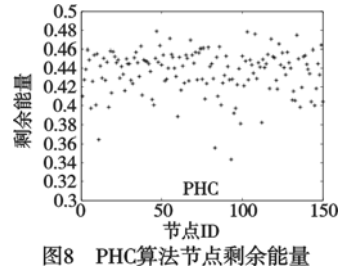

图8　PHC算法节点剩余能量

参考文献:

[1] MATEU L, MOLL F. Review of energy harvesting techniques and application for microelectronics[C]//Proc of SPIE. 2005:359-373.

[2] MOSER C, BRUNELLI D, THIELE L, *et al*. Real-time scheduling for energy harvesting sensor nodes[J]. Real-Time Systems, 2007, 37(3):233-260.

[3] RAGHUNATHAN V, KANSAL A, HSU J, *et al*. Design consideration for solar energy harvesting wireless embedded system[C]//Proc of the 4th Internatio-nal Symposium on Information Proceeding in Sensor Networks. Piscataway: IEEE Press. 2005:457-462.

[4] GAO Yi, SUN Gui-ling, LI Wei-xiang, *et al*. Wireless sensor node design based on solar energy supply[C]//Proc of the 2nd International Conference on Power Electronics and Intelligent Transportation System. 2009:203-207.

[5] 樊晓平,杨玺,刘少强,等. 具有能量补给的无线传感器网络分簇路由算法[J]. 计算机工程,2008,11(34):120-123.

(上接第 259 页)

[3] LEONG K S, NG M L, GRASSO A R, *et al*. Synchronization of RFID readers for dense RFID reader environments[C]//Proc of International Symposium on Applications on Internet Workshops. Washington DC: IEEE Computer Society, 2006:295-298.

[4] 余松森,詹宜巨,彭卫东. 基于返回式索引的二进制树形搜索反碰撞算法及其实现[J]. 单片机工程与应用,2004,40(16):26-28.

[5] 鞠伟成,俞承芳. 一种基于动态二进制的 RFID 抗冲突算法[J]. 复旦大学学报:自然科学版,2005,44(1):46-50.

[6] 姜丽芬,卢桂章,辛运帷. 射频识别系统中的防碰撞算法研究[J]. 计算机工程与应用,2007,43(15):29-32.

[7] 刘猛,徐展. RFID 中多标签识别缺陷与防碰撞算法分析[J]. 单片机与嵌入式系统应用,2010(1):18-20.

[8] 亢军贤,张瑞生,夏禹,等. 基于 RFID 的二进制防碰撞算法的改进[J]. 微计算机信息,2010(8):150-152.

[9] 陈冲,徐志,何明华. 一种新的 RFID 防碰撞算法的研究[J]. 福州大学学报:自然科学版,2009,37(3):367-371.

[10] 谢振华,赖声礼,陈鹏. 标签防冲撞算法设计[J]. 计算机工程,2008,34(6):90-92.