(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 104093184 B
(45)授权公告日 2017.08.25

(21)申请号 201410320566.2

(22)申请日 2014.07.07

(65)同一申请的已公布的文献号
申请公布号 CN 104093184 A

(43)申请公布日 2014.10.08

(73)专利权人 北京理工大学
地址 100081 北京市海淀区中关村南大街5号北京理工大学

(72)发明人 礼欣 李遵光 张鸿雁 樊秀梅

(74)专利代理机构 北京理工正阳知识产权代理事务所(普通合伙) 11639
代理人 唐华

(51)Int.Cl.
H04W 40/10(2009.01)
H04W 84/18(2009.01)

(56)对比文件
Hongyan Zhang 等.An optimal solution for round rotation time setting in LEACH.《International conference on wireless algorithms,systems and applications》.2013,

审查员 蒋莉

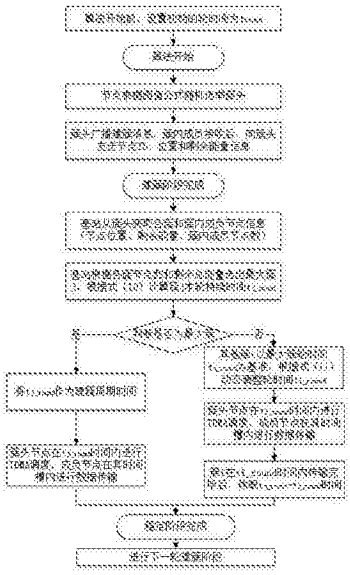
权利要求书1页 说明书6页 附图7页

(54)发明名称

一种基于能耗的LEACH轮换时间动态调整方案

(57)摘要

本发明涉及一种基于LEACH协议的轮换时间动态调整方案，属于无线传感器网络技术领域。该方案包括如下内容：簇形成后，簇头节点收集簇内所有成员节点的位置和剩余能量信息，并将信息融合后向基站回报；基站根据各簇的信息，评选出最大簇，并根据最大簇的剩余能量调整该轮的持续时间以及其它簇的数据传输时间。LEACH协议中应用本发明提出的基于能耗的LEACH协议的轮换时间动态调整方案，可以有效优化选簇周期、平衡节点间能量消耗、延长网络生存周期。



CN 104093184 B

1.一种基于LEACH协议的轮换时间动态调整方案,其特征在于,包括以下内容:

(一)网内各节点具备感知自己位置和剩余能量的能力;

(二)簇形成后,簇头节点收集簇内所有成员节点的位置和剩余能量信息;

(三)簇头节点向基站报告簇内所有成员节点的剩余能量或剩余总能量和节点数信息;

(四)基站根据收集到的所有簇信息,按照节点数和剩余总能量由大到小对簇排序,并将排在首位的簇选为最大簇,记最大簇的簇号为j;

(五)基站根据最大簇的剩余总能量使用下式动态调整最大簇的本轮持续时间:

$t_{j\_round}=t_{round}(E_{j\_current}/E_{j\_init})$

其中,t_{round}为基准轮时间,t_{j_round}为簇j在本轮的持续时间,$E_{j_current}$为簇j的剩余总能量,E_{j_init}为簇j的初始总能量;

使用下式动态调整其它簇在本轮的数据传输时间:

$$t_{i\_round}=\frac{E_{elec}\cdot n_j+E_{DA}\cdot n_j+\varepsilon_{amp}d_{j\_toBS}^4+\sum_{k=1}^{n_j-1}(E_{elec}+\varepsilon_{fs}d_{k\_toCH}^2)}{E_{elec}\cdot n_i+E_{DA}\cdot n_i+\varepsilon_{amp}d_{i\_toBS}^4+\sum_{k=1}^{n_i-1}(E_{elec}+\varepsilon_{fs}d_{k\_toCH}^2)}\cdot\left(\frac{E_{i\_current}}{E_{i\_init}}\right)\cdot\left(\frac{n_i}{n_j}\right)^2\cdot t_{j\_round}$$

其中,t_{i_round}为簇i在本轮的数据传输时间,t_{j_round}为簇j在本轮的持续时间,$E_{i_current}$为簇i的剩余总能量,E_{i_init}为簇i的初始总能量,n_i为簇i包含的节点数,n_j为簇j包含的节点数,E_{elec}表示传播一比特数据所消耗的能量,E_{DA}表示聚合一比特数据所需能量,d_{i_toBS}代表簇i的簇头节点到基站的距离,d_{k_toCH}代表簇内成员节点k到簇头结点的距离,ε_{amp}表示当$d\geq d_0$时射频放大器传输1bit单位平方米 的平方所耗的能量,d_0为预先假定的阈值距离,ε_{fs}表示当$d<d_0$时射频放大器传输1bit单位平方米所耗的能量;

(六)基站将t_{i_round}和t_{j_round}发回各簇的簇头节点;

(七)最大簇的簇头节点在t_{j_round}时间内为簇内各成员节点分配数据传输时隙,其它簇的簇头节点在t_{i_round}时间内为簇内各成员节点分配数据传输时隙;

(八)各簇内成员节点在簇头节点为其分配的时隙内向簇头传输数据,簇头节点对接收到的数据进行数据融合后再将信息传送给基站;

(九)各簇i在经过t_{i_round}时间后,休眠$t_{j_round}-t_{i_round}$时间后被唤醒,此后全网进入新一轮的簇建立和稳定运行阶段,簇形成后从步骤(二)重新开始新一轮的轮时间动态调整和稳定运行过程。

一种基于能耗的LEACH轮换时间动态调整方案

技术领域

[0001] 本发明涉及一种基于能耗的Leach轮换时间动态调整方案，属于无线传感器网络技术领域。

背景技术

[0002] 延长网络的生命周期是一个无线传感网络中重要的问题。很多学者也提出了很多高效节能的协议来延长无线传感网络的生命周期。目前，通过网络分簇来实现能量的均衡以此实现延长网络生命周期是研究的热点方向。LEACH算法是由Heinzelman等人提出的最流行的基于分簇的高效节能的通信协议，协议通过对节点进行分簇并将能量均衡负载到各个簇中来降低网络中总能量的消耗。为了确保能量的均衡消耗，LEACH周期性的在所有节点中随机选择节点来充当簇头。LEACH协议是以“轮”来实现的。每一轮包括：建立阶段和稳定运行阶段。建立阶段要完成簇头节点的选择、簇头节点的广播、非簇头节点的加入的簇形成过程以及簇头节点为簇内成员分配TDMA时隙的TDMA调度过程；在稳定运行阶段，簇头接受来自其成员的消息并将聚合后的数据发送到基站。然而，LEACH的许多参数可以影响协议的性能，这些参数都有待优化，如，阈值，簇的数目等等。因此，许多学者都致力于优化LEACH的参数来提高LEACH的性能。尽管很多人对LEACH做了不同的优化，但是LEACH每一轮持续的时间 却很少有人研究。每一轮持续时间的长短对于网络的整个性能是很关键的。如果时间太长，那么簇头就长时间的处于活跃状态，簇头节点的能量就会很快的消耗完毕；如果时间太短，簇头的选择过于频繁导致过多的能量浪费在簇建立阶段，因为这一阶段是不会发送数据的。

发明内容

[0003] 本发明的目的是为了克服已有技术存在的缺陷，提出了一种基于能耗的Leach轮换时间动态优化算法。

[0004] 本发明的思想是根据簇内节点数量和能量消耗的不同，动态的调整每轮的持续时间，以此来优化LEACH的选簇周期、平衡节点间的能量消耗、延长网络的生存周期。

[0005] 本发明的目的是通过以下技术方案实现的：

[0006] 一种基于能耗的LEACH协议的轮换时间动态调整方案，包括以下内容：

[0007] (一)网内各节点具备感知自己位置和剩余能量的能力；

[0008] (二)簇形成后，簇头节点收集簇内所有成员节点的位置和剩余能量信息；

[0009] (三)簇头节点向基站报告簇内所有成员节点的剩余能量或剩余总能量和节点数信息；

[0010] (四)基站根据收集到的所有簇信息，按照节点数和剩余总能量由大到小对簇排序，并将排在首位的簇选为最大簇，记最大簇的簇号为j；

[0011] (五)基站根据最大簇的剩余总能量使用下式动态调整最大 簇的本轮持续时间：

[0012] $t_{j\_round} = t_{round}(E_{j\_current}/E_{j\_init})$

[0013] 其中，t_{round}为基准轮时间，t_{j_round}为簇j在本轮的持续时间，$E_{j_current}$为簇j的剩余总能量，E_{j_init}为簇j的初始总能量；

[0014] 使用下式动态调整其它簇在本轮的数据传输时间：

[0015]

$$t_{i\_round}=\frac{E_{elec}\cdot n_j+E_{DA}\cdot n_j+\varepsilon_{amp}d_{j\_toBS}^4+\sum_{k=1}^{n_j-1}(E_{elec}+\varepsilon_{fs}d_{k\_toCH}^2)}{E_{elec}\cdot n_i+E_{DA}\cdot n_i+\varepsilon_{amp}d_{i\_toBS}^4+\sum_{k=1}^{n_i-1}(E_{elec}+\varepsilon_{fs}d_{k\_toCH}^2)}\cdot\left(\frac{E_{i\_current}}{E_{i\_init}}\right)\cdot\left(\frac{n_i}{n_j}\right)^2\cdot t_{j\_round}$$

[0016] 其中，t_{i_round}为簇i在本轮的数据传输时间，t_{j_round}为簇j在本轮的持续时间，$E_{i_current}$为簇i的剩余总能量，E_{i_init}为簇i的初始总能量，n_i为簇i包含的节点数，n_j为簇j包含的节点数，E_{elec}表示传播一比特数据所消耗的能量，E_{DA}表示聚合一比特数据所需能量，d_{i_toBS}代表簇i的簇头节点到基站的距离，d_{k_toCH}代表簇内成员节点k到簇头结点的距离，ε_{amp}表示当$d\geq d_0$时射频放大器传输1bit单位平方米的平方所耗的能量，d_0为预先假定的阈值距离，ε_{fs}表示当$d<d_0$时射频放大器传输1bit单位平方米所耗的能量；

[0017] (六)基站将t_{i_round}和t_{j_round}发回各簇的簇头节点；

[0018] (七)最大簇的簇头节点在t_{j_round}时间内为簇内各成员节点分配数据传输时隙，其它簇的簇头节点在t_{i_round}时间内为簇内各成员节点分配数据传输时隙；

[0019] (八)各簇内成员节点在簇头节点为其分配的时隙内向簇头传输数据，簇头节点对接收到的数据进行数据融合后再将信息传送给 基站；

[0020] (九)各簇i在经过t_{i_round}时间后，休眠$t_{j_round}-t_{i_round}$时间后被唤醒，此后全网进入新一轮的簇建立和稳定运行阶段，簇形成后从步骤(二)重新开始新一轮的轮时间动态调整和稳定运行过程。

[0021] 有益效果

[0022] 对比传统的LEACH协议，本发明方案可以有效地均衡网内节点的能量消耗，延长网络生存周期以及基站接收信息包的数量；而且，该方案适用于目前提出的轮时间固定的所有基本LEACH协议及其改进协议，适用范围广泛。

附图说明

[0023] 图1为LEACH协议拓扑结构图。

[0024] 图2为改进的LEACH算法流程图。

[0025] 图3为50、100、200个节点的场景中，不同轮换时间的情况下生存周期的比较图。

[0026] 图4为在不同轮换时间下，100个节点场景的节点生存时间的对比图。

[0027] 图5为在不同轮换时间下，100个节点场景的能量消耗速率的对比图。

[0028] 图6为在不同轮换时间下，100个节点场景的基站接收数据量的对比图。

[0029] 图7为轮换时间为20s时，改进前后的LEACH算法在网络生存 周期上的对比图。

[0030] 图8为轮换时间为20s时，改进前后的LEACH算法在节点生存时间上的对比图。

[0031] 图9为轮换时间为20s时，改进前后的LEACH算法在节点平均能量消耗上的对比图。

[0032] 图10为轮换时间为20s时，改进前后的LEACH算法在基站接收数据量上的对比图。

[0033] 图11为轮换时间为20s时，改进前后的LEACH算法在第一个节点死亡时间和一半节点死亡时间上的对比图。

具体实施方式

[0034] 下面结合附图和实施例对本发明做进一步说明。

[0035] 本发明的LEACH协议拓扑图如图1所示，各簇头节点与基站相互通信，各簇内节点与本簇内簇头节点相互通信。下面介绍本发明的实施过程以及评估结果。

[0036] 步骤一、根据本发明方案修改原有LEACH协议代码：

[0037] 如图2所示，对原LEACH协议作出改进。改进以后，我们可以在网络模拟平台NS2上对改进的LEACH算法进行仿真实验。

[0038] 改进后的LEACH路由协议在一轮中的工作流程为：

[0039] 1.算法开始前，设置初始的轮时间为t_{round}；

[0040] 2.每个节点随机选择0-1之间的一个值，如果选定的值小于某一个阈值，那么这个节点成为簇头节点；选定簇头节点后，通过广播告知整个网络；网络中的其他节点根据接收信息的信号强度决定从属的 簇，并向其发送加入信息，此信息包含节点ID、节点位置和节点剩余能量；

[0041] 3.簇头节点向基站报告簇内所有成员节点的剩余总能量和节点数信息；

[0042] 4.基站根据收集到的所有簇信息，按照节点数和剩余总能量由大到小对簇排序，并将排在首位的簇选为最大簇，记最大簇的簇号为j；

[0043] 5.基站根据最大簇的剩余总能量使用下式动态调整最大簇的本轮持续时间：

[0044] $t_{j\_round}=t_{round}(E_{j\_current}/E_{j\_init})$

[0045] 其中，t_{round}为基准轮时间，t_{j_round}为簇j在本轮的持续时间，$E_{j_current}$为簇j的剩余总能量，E_{j_init}为簇j的初始总能量；

[0046] 使用下式动态调整其它簇在本轮的数据传输时间：

[0047]
$$t_{i\_round}=\frac{E_{elec}\cdot n_j+E_{DA}\cdot n_j+\varepsilon_{amp}d_{j\_toBS}^4+\sum_{k=1}^{n_j-1}(E_{elec}+\varepsilon_{fs}d_{k\_toCH}^2)}{E_{elec}\cdot n_i+E_{DA}\cdot n_i+\varepsilon_{amp}d_{i\_toBS}^4+\sum_{k=1}^{n_i-1}(E_{elec}+\varepsilon_{fs}d_{k\_toCH}^2)}\cdot\left(\frac{E_{i\_current}}{E_{i\_init}}\right)\cdot\left(\frac{n_i}{n_j}\right)^2\cdot t_{j\_round}$$

[0048] 其中，t_{i_round}为簇i在本轮的数据传输时间，t_{j_round}为簇j在本轮的持续时间，$E_{i_current}$为簇i的剩余总能量，E_{i_init}为簇i的初始总能量，n_i为簇i包含的节点数，n_j为簇j包含的节点数，E_{elec}表示传播一比特数据所消耗的能量，E_{DA}表示聚合一比特数据所需能量，d_{i_toBS}代表簇i的簇头节点到基站的距离，d_{k_toCH}代表簇内成员节点k到簇头结点的距离，ε_{amp}表示当$d\geq d_0$时射频放大器传输1bit单位平方米的平方所耗 的能量，d_0为预先假定的阈值距离，ε_{fs}表示当$d<d_0$时射频放大器传输1bit单位平方米所耗的能量；

[0049] 6.基站将t_{i_round}和t_{j_round}发回各簇的簇头节点；

[0050] 7.最大簇的簇头节点在t_{j_round}时间内进行TDMA调度，为簇内各成员节点分配数据传输时隙；其它簇的簇头节点在t_{i_round}时间内进行TDMA调度，为簇内各成员节点分配数据

传输时隙；

[0051] 8.各簇内成员节点在簇头节点为其分配的时隙内向簇头传输数据，簇头节点对接收到的数据进行数据融合后再将信息传送给基站；

[0052] 9.各簇i在经过t_{i_round}时间后，休眠$t_{j_round}-t_{i_round}$时间后被唤醒；此后全网进入新一轮的簇建立和稳定运行阶段，转到步骤2重新开始，直到全网的剩余节点个数小于既定值k。

[0053] 步骤二、设置相应的实验参数：

[0054] 本发明试验场景为：在100×100m的范围内，随机分布50、100或200个节点，基站位于(50,175)的位置。基站的初始能量是无限的，普通节点的初始能量为2mJ，并且后续不能供应。无线电接收数据消耗能量E_{elec}设置为50nJ/bit，数据融合所需计算能量设置为5nJ/bit，簇头比例N/k为5％，数据包大小l为4000bits，比特率R_b为1Mbps，射频放大器的能量ε_{amp}和ε_{fs}分别为0.0013pJ/bit/m^4和10pJ/bit/m^2。

[0055] 步骤三、运行协议，分析其性能：

[0056] 参照图3，得出不同轮换时间下LEACH改进算法的网络生命周期。从图中可以看出，在相同的网络配置的情况下，节点的数量不同， 网络生命周期的峰值出现的位置就不同，并且节点少的峰值出现的早，节点多的峰值出现的晚。这是因为随着节点数量的增多，产生含簇内节点多的簇的概率就会增大，按照本文提出的动态调整每轮持续时间的算法，应该适当增加初始建簇时间，以实现节点能量消耗均匀。从图中可以看出，峰值分别出现在16s、20s和26s。

[0057] 参照图4，得出不同初始轮换时间的情况下，网络中节点的生存情况。可以看出，在初始建簇时间设置为20s时，节点死亡速度最慢，从图中也可以看出，即使初始建簇时间有所改变，但是第一个死亡节点的出现时间相差不是很大，这是因为，采用动态轮时间的LEACH协议会根据节点的剩余和消耗的能量来动态调整每轮持续时间，在经过若干轮后，轮时间会调整到当前能量环境下比较合适的水平，这会均衡节点的能量消耗，延后了第一个死亡节点的产生时间。

[0058] 参照图5，得出100个节点网络能量消耗的情况，从图中可以看出，当初始建簇时间设置为20s时，能量消耗的最慢。而且还能看出当网络能量非常低的时候，网络还能继续运行较长一段时间，这是因为采取了动态改变每轮时间的方法，当能量减少到一定程度，每轮的持续时间已经减少了很多，这就导致节点能量会减少缓慢，因此在网络能量消耗殆尽的时候，还能运行较长一段时间。

[0059] 参照图6，得出在不同初始建簇时间的情况下，基站接收到的数据包数量的比较，从图中可以看出，在初始建簇时间为20s时基站接收到的数据包数量最多。如果每轮持续时间设置的过长，簇头结点能量消耗过快而死亡，那么簇头节点处，聚合的数据就会变少，从而基 站收到的数据就会变少。如果每轮持续时间过短，那么会有过多的能量浪费在选簇阶段，而这一阶段中，节点根本不会发送数据，从而导致基站收到数据减少。

[0060] 步骤四、比较分析改进前后的LEACH算法的性能：

[0061] 从原LEACH协议的仿真结果中可以发现，在100个节点的场景中，原LEACH协议在初始轮换时间为20s时，网络生存周期最长。因此有必要对两者在促使轮换时间为20s时，做一下性能对比分析。为了方便进行对比，我们把优化前的算法称作S-LEACH(Static LEACH)算

法，优化后的算法称作D-LEACH(Dynamic LEACH)我们主要是对节点的生命周期、节点的能量消耗、基站接收的数据量、第一个节点和一半节点的死亡时间等进行分析。

[0062] 参照图7，S-LEACH和D-LEACH在网络生命周期上的对比图，可以看出，D-LEACH算法在所设置的各种初始建簇时间下，网络生命周期都比S-LEACH算法有较大的改进，并且在100个节点的情况下，两者的峰值都出现在建簇时间为20s的时候。

[0063] 参照图8，S-LEACH和D-LEACH在节点死亡速率上的比较图，可以看出，D-LEACH的生命周期要比S-LEACH长40％左右，而且在出现第一个节点死亡的时候，S-LEACH在大概200轮后网络完全死亡，但是D-LEACH却用了250轮，这是因为D-LEACH的每轮持续时间是随着簇内剩余能量和簇内节点个数动态调整的，随着节点的不断死亡，簇内节点的个数会相对减少，每轮的持续时间也会相对下降，这样就可以让节点的能量减少的相对较慢，因此在出现节点死亡 的时候D-LEACH还能比S-LEACH多持续一段时间。

[0064] 参照图9，S-LEACH和D-LEACH在平均能量消耗上的对比图，可以发现，D-LEACH的能量消耗比S-LEACH要慢，S-LEACH的斜率在整个网络生存周期中基本上不变，但是D-LEACH在510s前斜率是不断降低的，这是由于随着节点的能量消耗，每轮持续时间有所下降，这必然会降低能量消耗，但是在510s到600s之间，D-LEACH的能量消耗突然加快，这是因为，D-LEACH在510s时，第一个节点死亡，而D-LEACH的阈值选取公式是基于剩余能量的，随着时间的推进，节点剩余能量会比较低，造成节点的阈值比较小，节点被选为簇头的概率就比较小，在没有节点死亡的时候，这个后果还不是很明显，但是当有节点死亡的时候，能被选为簇头节点的数量就更少，当一个网络中没有节点当选簇头，或者节点没有加入到任何簇中，节点都会直接向基站发送信息，这个过程会消耗大量的能量。因此会造成能量消耗增加的比较快。而在600s的时候，节点的能量消耗会慢下来，这是因为，节点的平均剩余能量比较低，导致每轮的持续时间很短，节点每轮消耗的能量就很少，因此还能持续较长一段时间。

[0065] 参照图10，S-LEACH和D-LEACH在基站接收到的数据量方面的比较图，由图中可知，D-LEACH接收到的数据量和S-LEACH相比基本持平，但是数据量的增长率要小一些，这是因为，D-LEACH为了让节点能量均衡消耗，含节点少的簇的每轮持续时间就会少一些，发送的数据量就会相应减少，这会导致整个网络每轮发送的数据总量比S-LEACH要少，并且随着时间的推移，D-LEACH的斜率慢慢降 低，到大约640s时，数据量不再增长，虽然网络还在继续运行，但是没有起到应有的作用，因此我们可以认为在当前情况下，D-LEACH的生命周期为640s。

[0066] 参照图11，S-LEACH和D-LEACH在第一个节点死亡时间和一半节点死亡时间两方面的比较图，可以看出，D-LEACH的第一个节点死亡的时间比S-LEACH晚很多，这是因为D-LEACH算法均衡了节点能量的消耗，这会大大延迟第一个节点死亡的时间。在一半节点死亡时间的对比下，D-LEACH也要优于S-LEACH，但是也可以发现，D-LEACH的HNA和FND之间相差的较少，这是因为由于能量均衡消耗，在出现第一个节点死亡的时候，其他节点的能量水平也已经降低很多，因此部分节点死亡的比S-LEACH要快。

[0067] 综上所述，本发明在多项性能指标上优于原轮时间固定的LEACH协议。

[0068] 以上所述的具体描述，对发明的目的、技术方案和有益效果进行了进一步详细说明，所应理解的是，以上所述仅为本发明的具体实施例而已，并不用于限定本发明的保护范围，凡在本发明的精神和原则之内，所做的任何修改、等同替换、改进等，均应包含在本发明

的保护范围之内。

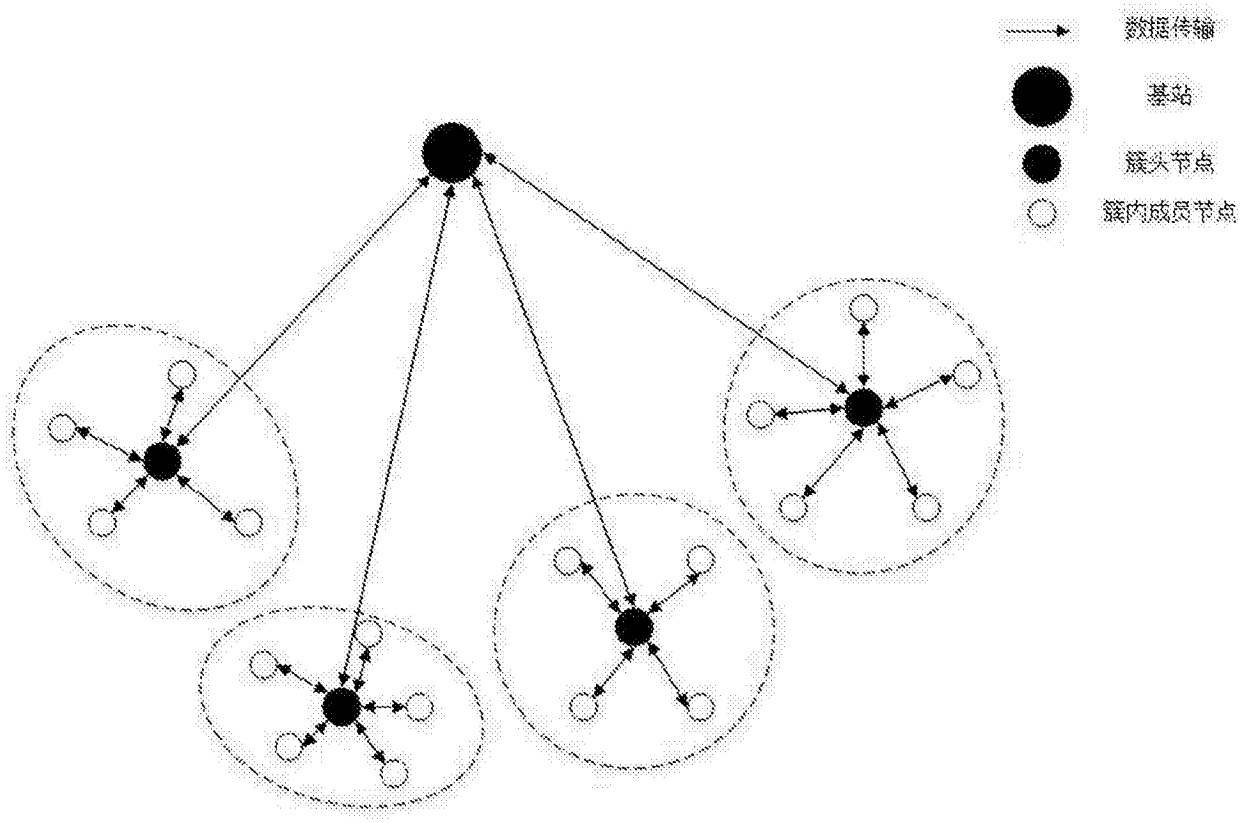


图1

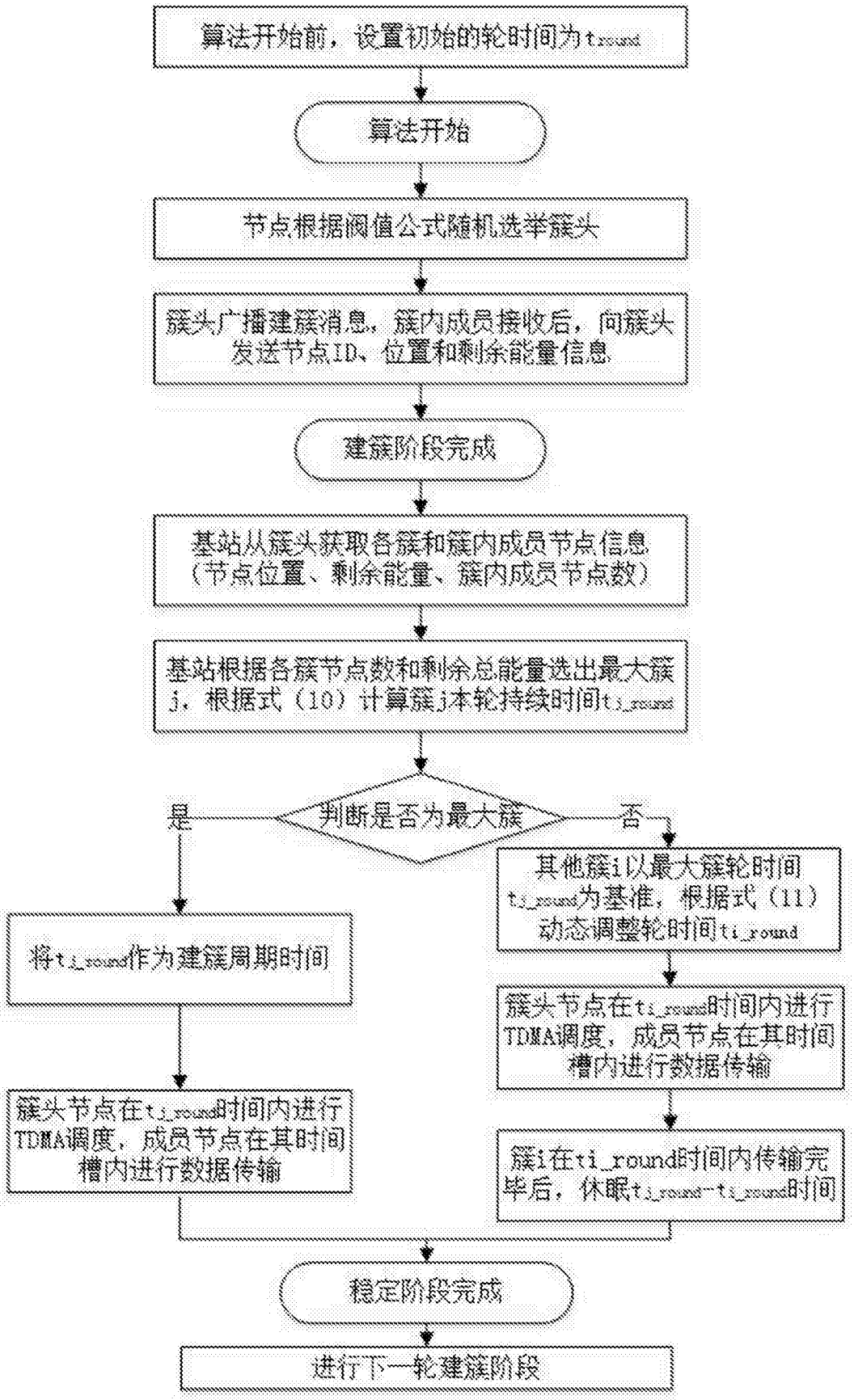

算法开始前，设置初始的轮时间为t_{round}
算法开始
节点根据阈值公式随机选举簇头
簇头广播建簇消息，簇内成员接收后，向簇头发送节点ID、位置和剩余能量信息
建簇阶段完成
基站从簇头获取各簇和簇内成员节点信息（节点位置、剩余能量、簇内成员节点数）
基站根据各簇节点数和剩余总能量选出最大簇j，根据式（10）计算簇j本轮持续时间t_{j_round}
判断是否为最大簇
是
否
将t_{j_round}作为建簇周期时间
簇头节点在t_{j_round}时间内进行TDMA调度，成员节点在其时间槽内进行数据传输
其他簇i以最大簇轮时间t_{j_round}为基准，根据式（11）动态调整轮时间t_{i_round}
簇头节点在t_{i_round}时间内进行TDMA调度，成员节点在其时间槽内进行数据传输
簇i在t_{i_round}时间内传输完毕后，休眠$t_{j_round}-t_{i_round}$时间
稳定阶段完成
进行下一轮建簇阶段

图2

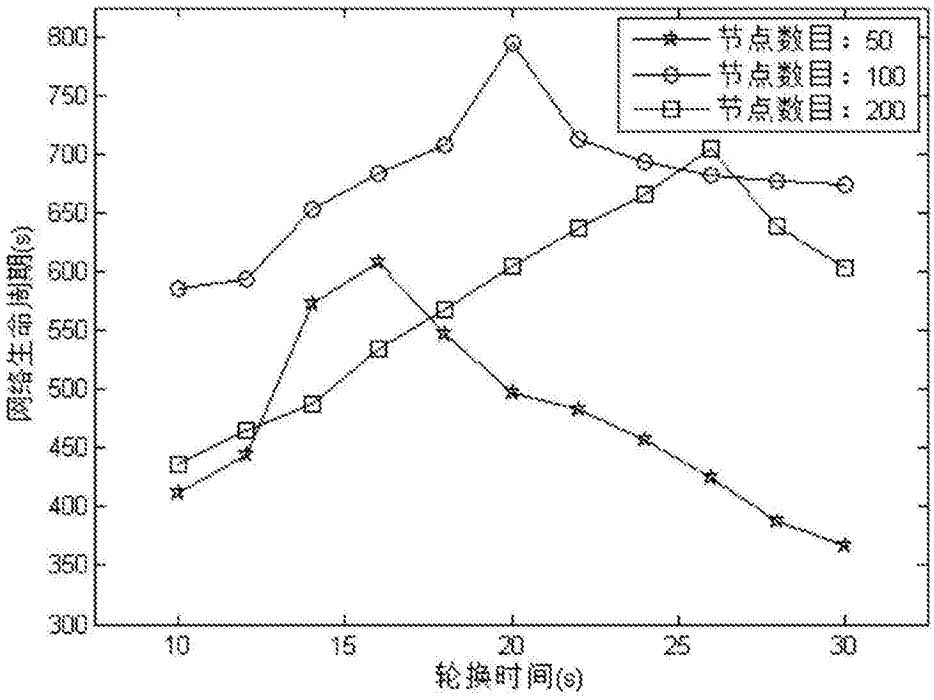

节点数目：50
节点数目：100
节点数目：200
网络生命周期(s)
轮换时间(s)

图3

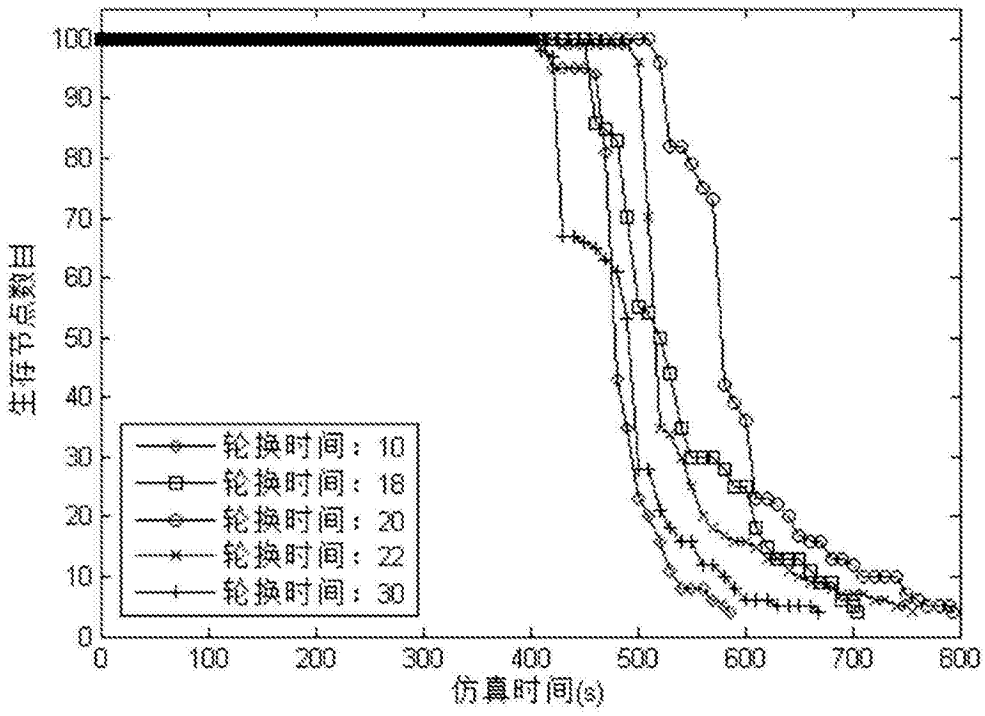

轮换时间：10
轮换时间：16
轮换时间：20
轮换时间：22
轮换时间：30
存活节点数目
仿真时间(s)

图4

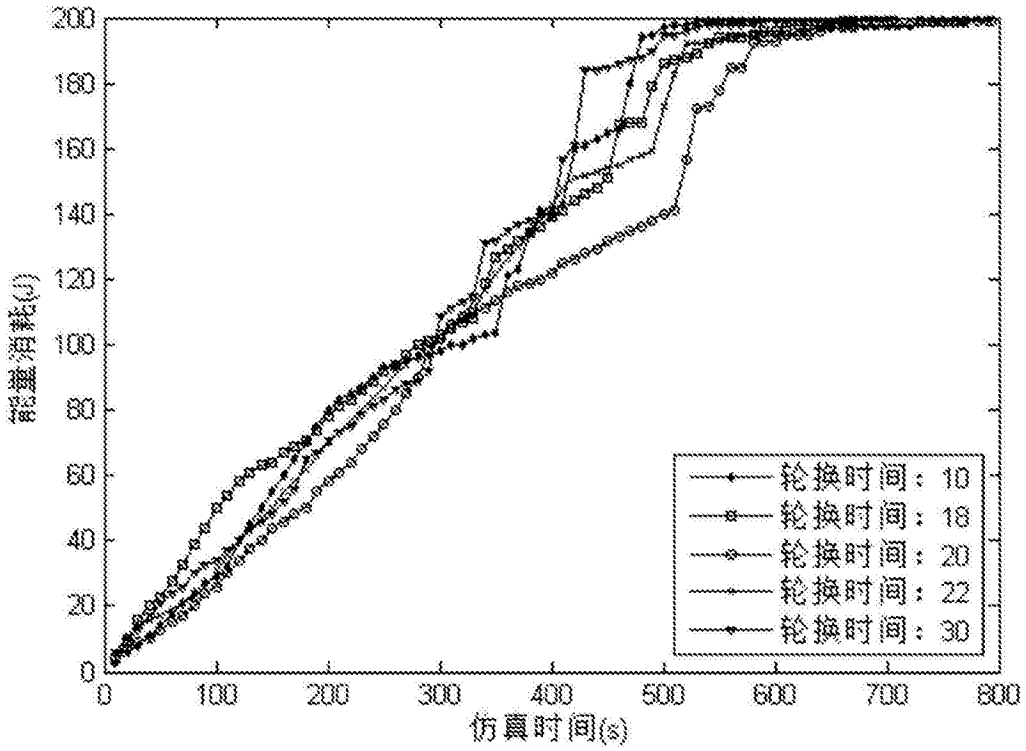


图5

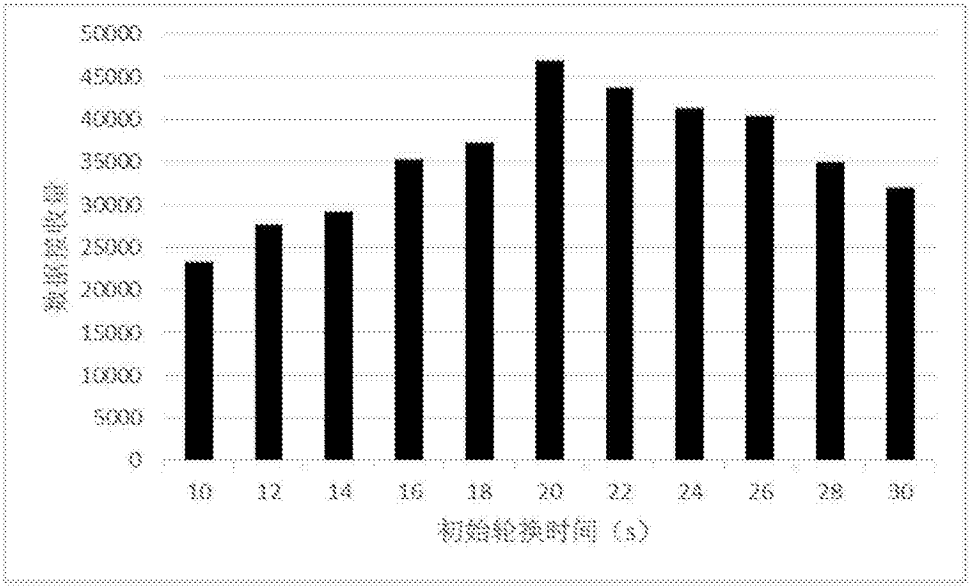


图6

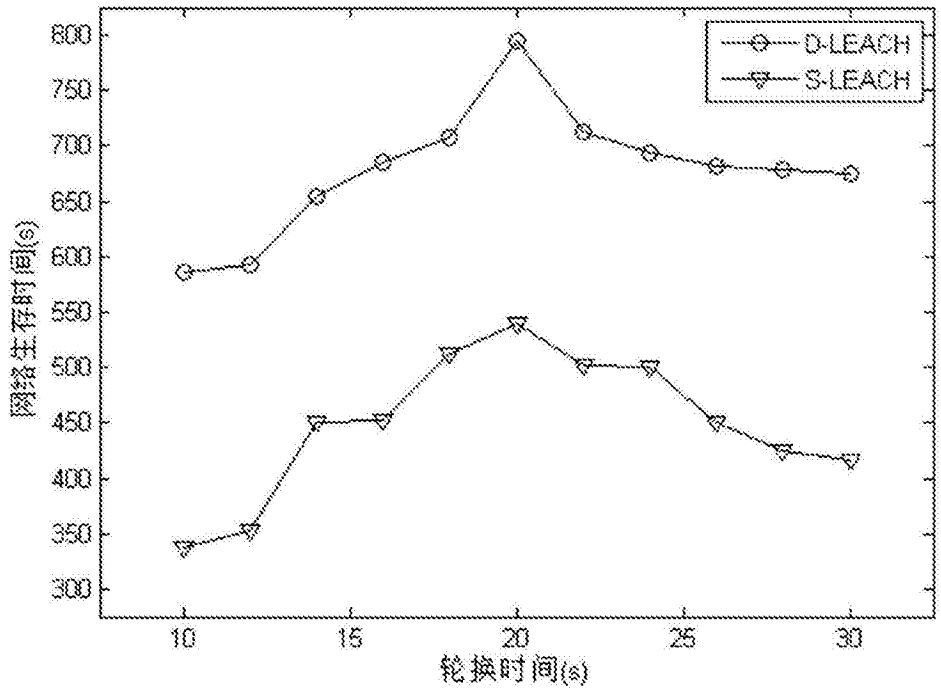

D-LEACH
S-LEACH
网络生存时间(s)
轮换时间(s)
800
750
700
650
600
550
500
450
400
350
300
10
15
20
25
30

图7

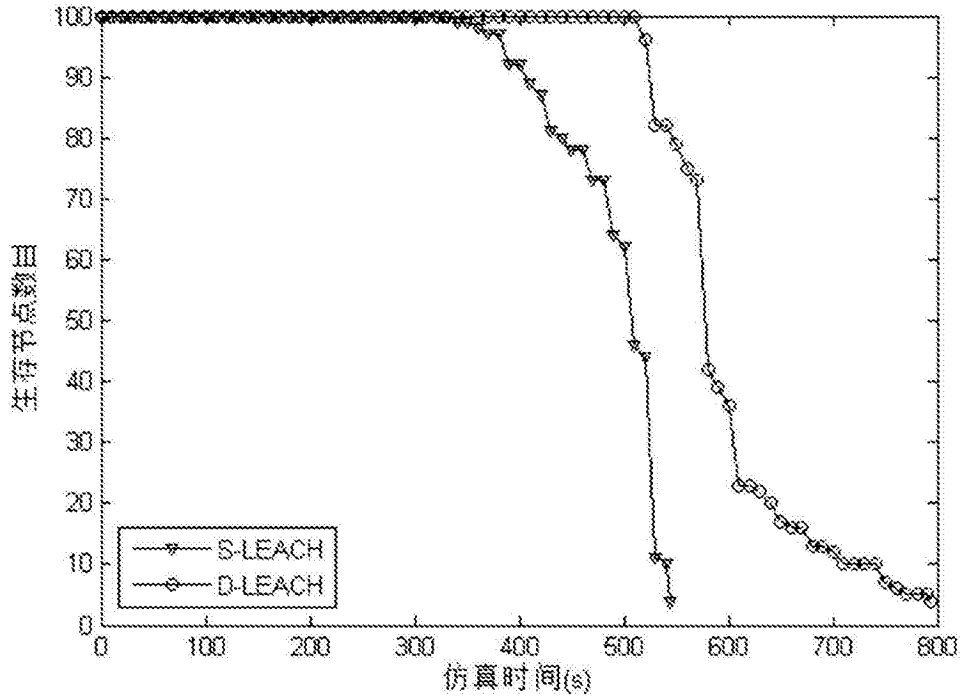

S-LEACH
D-LEACH
生存节点数目
仿真时间(s)
100
90
80
70
60
50
40
30
20
10
0
0
100
200
300
400
500
600
700
800

图8

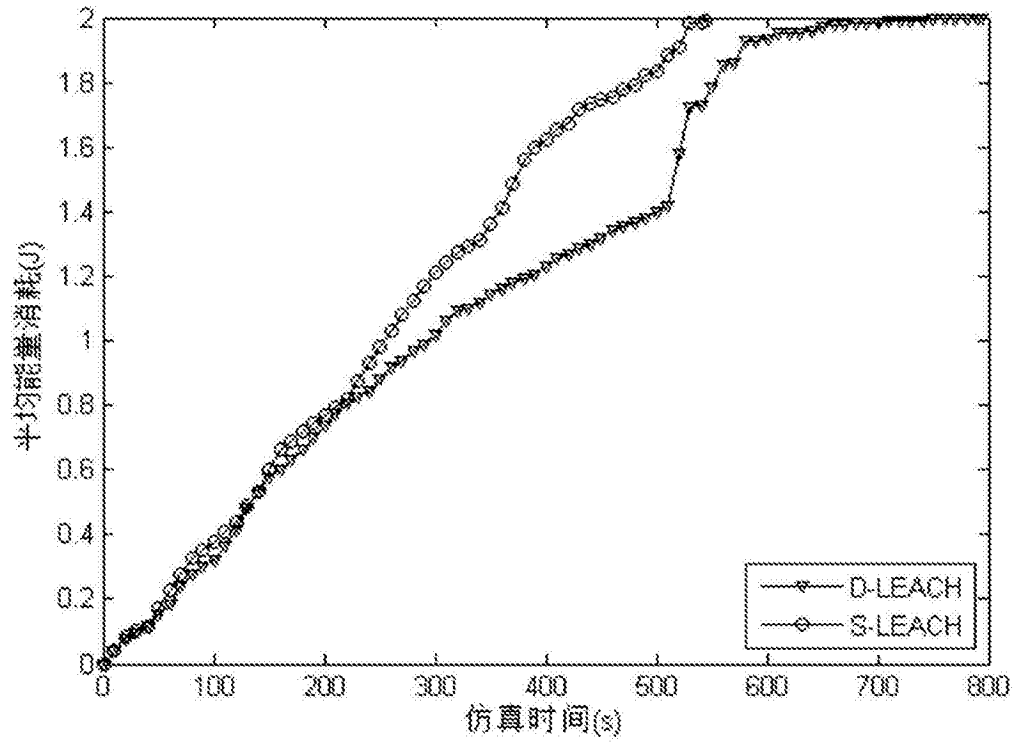

平均能量消耗(J)
仿真时间(s)
D-LEACH
S-LEACH

图9

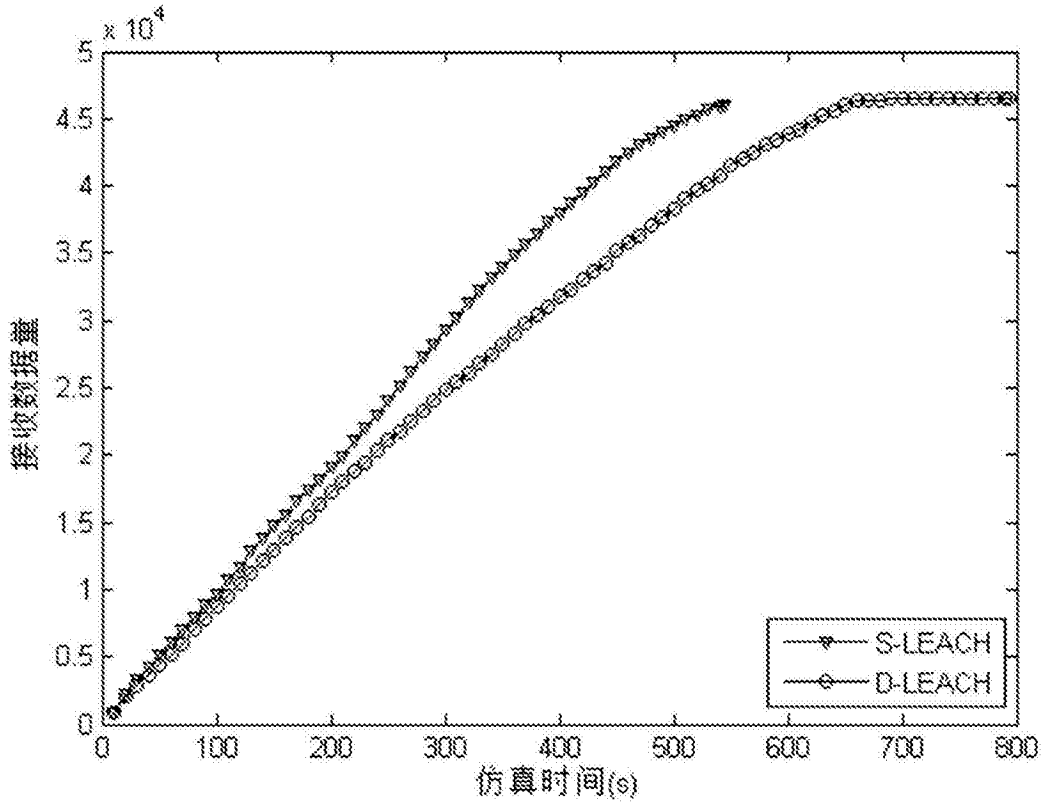

x 10^4
接收数据量
仿真时间(s)
S-LEACH
D-LEACH

图10

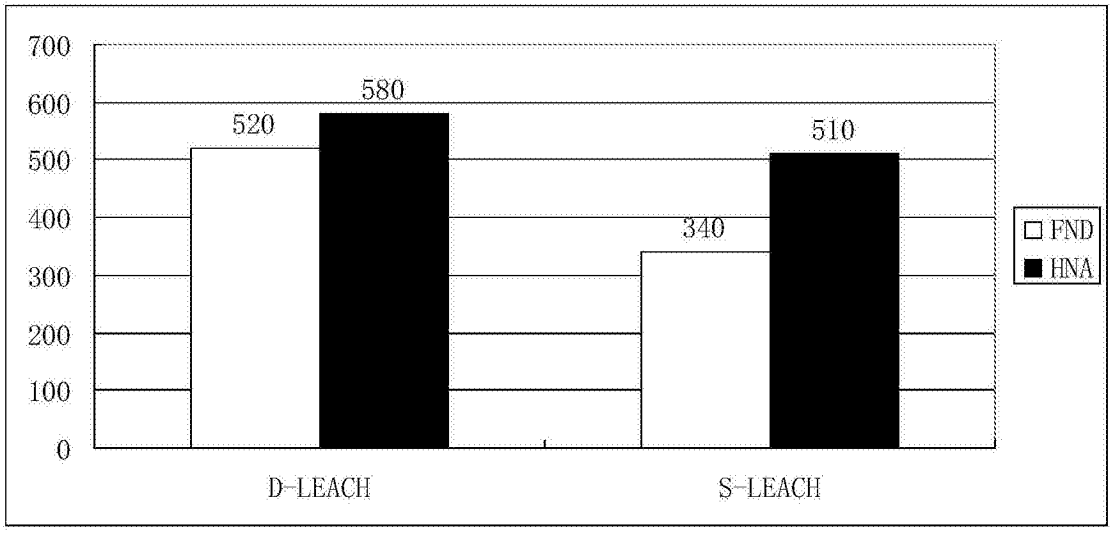

700
600
500
400
300
200
100
0
520
580
340
510
D-LEACH
S-LEACH
FND
HNA

图11