无线传感器网络中基于剩余能量的联合选举动态成簇路由算法

郭 彬　　李 喆
(东北大学信息科学与工程学院　沈阳　110004)

摘　要：针对无线传感器网络中突发事件监测等响应式网络应用，提出了一种基于能量的联合选举动态成簇算法。基于节点剩余能量，在事件区域内周期性地进行簇首选举，建立以簇首为根的簇树结构对事件区域内的数据进行搜集融合，从而减少网络中传输的数据量。仿真结果表明：该算法降低了节点平均能耗，具有良好的能量均衡效果，延长了网络生存时间。

关键词：无线传感器网络；动态成簇；剩余能量；能量均衡
中图分类号：TP393.06　　**文献标识码**：A　　**文章编号**：1009-5896(2007)12-3006-05

United Voting Dynamic Cluster Routing Algorithm Based on Residual-Energy in Wireless Sensor Networks

Guo Bin　　Li Zhe
(*College of Information Science and Engineering*, *Northeastern University*, *Shenyang* 110004, *China*)

Abstract: For monitoring burst events in reactive wireless sensor networks, a new United Voting Dynamic Cluster (UVDC) routing algorithm based on residual-energy is proposed. Cluster head is periodically selected according to residual-energy among the nodes located in the event area, meanwhile a structure of cluster tree rooted at the cluster head is formed within the event area to gather and aggregate data. The data transmitted in network is reduced. Simulation results show that UVDC decreases the average energy consumption, as well as balances the energy dissipation effectively, and extends the lifetime of network.

Key words: Wireless sensor network; Dynamic cluster; Residual-energy; Energy balance

1　引言

在无线传感器网络中由于传感器节点能量受限，因此在设计传感器路由协议的时候，要将节点能量放在考虑的第一位[1]。成簇的路由算法可以将多个节点的监测数据融合在一起，从而减少网络中传输的数据量，降低节点的能耗。LEACH[2]、TEEN[3]、APTEEN[4]等采用成簇路由算法，取得了一定的节能效果。但上述算法并不适合突发事件区域监测等响应式网络应用。(1) LEACH、TEEN等算法都采用节点根据平均概率轮流担当簇首的方法。通过分析发现，在响应式网络中，在某个节点担当簇首期间可能事件频发，造成簇首能量过度消耗；或者在节点担当簇首期间没有事件发生，节点能量被保留下来。LEACH、TEEN中采用的成簇算法无法区别这两种情况，因而无法完成节点间的能量均衡。(2) 对于突发事件区域监测等响应式网络的应用，当事件被发现后，一般只需要事件区域内的节点进行数据采集并传送出去，因此只需事件区域内节点进行成簇就可以完成对该事件的监测任务。LEACH、TEEN等协议中采用的成簇算法每次进行簇重构时全网节点都参与成簇工作，造成了事件区域外节点不必要的能耗。(3) 由于成簇和监测事件的无关联性，很容易造成事件区域与成簇结构不吻合，使同一事件区域内的节点不在同一个簇内，很可能分布在几个簇内。这样降低了数据搜集和融合的效率，也不利于用户对监测事件的分析。

因此，针对局部突发事件区域监测等的应用，最好使用动态成簇的方法。动态成簇算法中由监测事件动态触发成簇过程，整个事件区域内节点的信息汇聚到一个节点进行数据融合。事件区域外的节点则不必参与到这些工作中来，有效地节省了能量。在文献[5,6]中提出了两种动态成簇的算法。在文献[5]中网络节点分为能量充足的高端节点和其他低端节点。成簇时由监测信号最强的高端节点充当簇首，向周围节点搜集监测信息。而在文献[6]中从事件定位角度出发，通过事件区域内节点竞争，由监测信号强度最大的节点作为簇首来搜集数据。上述两种算法都是未考虑能量因素的动态成簇，文献[5]中需要网络中高端节点充当簇首，并且高端节点必须均匀分布在网络中；文献[6]中根据其簇首选举算法，如果监测信号强度最大的节点能量很少，它也会取得簇首的位置，那么将对这个节点能量造成更大的消耗。

本文针对突发事件区域监测等响应式应用，提出了一种新的基于节点能量的联合选举动态成簇路由(United Voting Dynamic Cluster, UVDC)算法。UVDC 通过事件区域内节点基于自身剩余能量的联合投票选举簇首节点，建立以簇首

2006-05-23 收到，2006-11-20 改回
国家 863 项目(2002AA784030)资助项目

为根的簇树网络对事件区域数据进行搜集和融合。

2　无传感器网络的能量均衡问题分析

对于能量相关的无线传感器网络，能耗均衡对于延长整个网络的生存时间至关重要[7]。如图 1 所示，使用NS2 仿真的 100 个节点的响应式网络，节点均匀分布在 100m×100m 的区域内，网络采用非能量相关的路由算法，随机产生一些突发监测事件，当网络运行一段时间后形成节点剩余能量的曲面图。深色区域表示节点的能量状况良好，而浅色表示节点的剩余能量较低。可以观察到，曲面凹凸不平，说明此时节点剩余能量并不均匀，非能量相关路由算法造成个别节点能量的过度消耗，缩短整个网络的生存时间。

在成簇算法中，一般情况下簇首的能耗要高于普通节点。因此成簇时应充分考虑节点的剩余能量，选取能量较高的节点充当簇首，使数据流向能量较高的区域。 通过这种策略使剩余能量曲面更趋于平坦，实现节点间的能耗均衡。基于这种思想设计了基于能量的联合选举动态成簇路由算法。

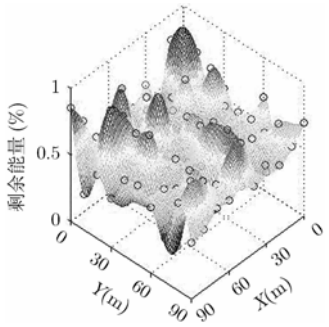


图 1 剩余能量分布曲面图

3　基能量的联合选举动态成簇路由算法

联合选举成簇算法中在监测事件的驱动下节点依据剩余能量通过相互投票动态选举簇首节点，在算法中作如下规定：(1) 只有节点位于事件区域内，即当监测值超过节点预先设定的感应阈值，节点才拥有选举权。(2)选举的规则是具有选举权的节点只会给那些比自己剩余能量大的邻居节点投票。(3)选票的传播距离只有一跳，不被转发，即一个节点只能给它的邻居节点投票。

联合选举成簇算法的成簇过程分为两个阶段：投票选举阶段和成簇阶段。在事件发生后触发投票选举过程，事件区域内节点依据上面规则进行联合投票。根据投票规则节点的剩余能量越少，投出的票就越多；反之，节点的剩余能量越多，它获得选票的可能性就越大。簇首的选择在考虑节点剩余能量的同时，还考虑了事件区域的拓扑结构的影响。当一个节点周围邻居比较多，而与邻居相比，本节点的剩余能量也占有一定的优势，那么本节点就会获得较多的选票。反之如果节点邻居很少，即使其剩余能量很大，也得不到很多的选票。因为这样的节点通常处于事件区域边缘或者死角，不适合担当簇首。

当投票过程结束后，依据式(1)每个节点都累积了一定的选票。得票最多的节点通过接下来的成簇过程成为本区域的簇首 CH(D)。

$$\mathrm{vote}(m)=\sum_{i\in N_m\in D}\mathrm{if}(E_m>E_i) \tag{1}$$

$$\mathrm{CH}(D)=\underset{m\in D}{\mathrm{Max}}\,\mathrm{vote}(m) \tag{2}$$

其中D是事件区域节点集合，CH(D)为D区域的簇首，N_m为节点m的邻居集合，E_m为节点m的剩余能量。

在成簇过程中，得票数超过阈值的节点参与对簇首的竞争。这个阈值的大小根据网络规模和节点密度而定。参与竞争的节点广播成簇请求，请求中包括本身得票数，剩余能量等信息。成簇请求有两个作用：

(1) 通过成簇请求的竞争扩散，事件区域内节点获取得票最多节点的状态信息，将这个节点设置为簇首节点。

(2) 成簇请求扩散的同时，事件区域内的节点建立起以簇首为根的簇树网络，用来搜集事件区域内的数据。而不是使用LEACH、TEEN中簇成员与簇首单跳的通信方式。这是由于簇规模依据事件区域大小动态变化，而单跳通信不能适应这种动态性。

为了减小簇内数据搜集的延迟时间，在簇内建立路由时采用了最小跳数算法(MHR)[8]构建生成树。同时增加了在最小跳数基础上对节点的剩余能量的考虑：在跳数相同的情况下，节点优先选择剩余能量较多的节点作为自己的父节点。

突发事件产生后，事件区域通过定期的联合选举动态成簇。剩余能量较多并且数据搜集效率较高的节点轮流成为簇首，建立簇树网络执行数据的搜集和融合任务。

3.1　算法描述

3.1.1 投票过程　投票选举阶段事件区域内节点根据当前剩余能量进行相互投票，累积一定数目的选票，为成簇阶段作准备。

算法设定：

(1) 为了减少能量消耗，通过设定节点的发送功率使节点只能与适当数目的一跳邻居通信。

(2) 各个节点设定监测阈值(inspectThreshold)，当节点的监测值超过这个门限，则认为本节点在事件区域内，节点具有选举权。监测阈值的设置与TEEN[3]中相同。

(3) 协议周期(protocol period)：当一个协议周期结束，事件区域进行下一轮新的动态成簇过程。每个协议周期亦称为一轮(round)。

(4) 节点自身状态数据结构如表 1。

表 1　节点状态数据结构

变量	意义
myid	节点标识
myvote	节点累积的选票数
myenergy	节点当前的剩余能量
myhoptosink	节点到Sink的跳数

(5) 选票报文(Vote Message)是一个短报文，类似AODV协议中的hello报文，其中只包括节点id和自身能量信息。投票过程使用伪代码描述如下：

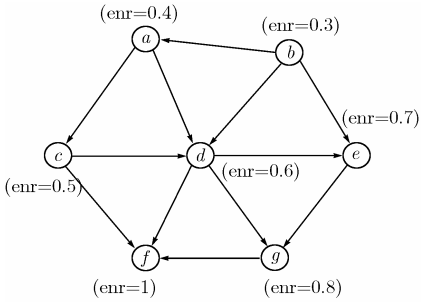
新协议周期开始：

```
Begin
If ( 节点监测值 > inspectThreshold )
{
    p→id=myid;
    p→type=选票报文;
    p→energy=myenergy;
    broadcast(p);
}
End
```

邻居节点接收选票报文：

```
Begin
If (节点监测值>inspectThreshold&&本身能量≥选票p中的能量)
{
    myvote++;
}
drop (p)
End
```

如图 2 ，a–g 节点分布在事件区域内，节点剩余能量表示为(0，100%)之间的随机值。根据上面算法，选举关系如图 2 所示。



enr: 节点剩余能量

图 2 节点选举关系

可以看出节点 d、节点 f 得票都为 3 票，但由于节点 f 能量优于节点 d，因此在本轮中节点 f 获得簇首位置。但并不是能量最多的节点始终能当选为簇首。例如把节点 d 能量改变为 0.8，根据选举规则，其得票数变为 5，节点 d 将以绝对优势当选为本轮簇首。因此在 UVDC 中簇首的选择除了与节点剩余能量有关外，还与事件区域拓扑相关。

3.1.2 成簇过程 事件区域节点投票结束后，进入到成簇阶段，其中部分节点发出成簇请求，竞争成为簇首，并且建立以簇首为根的簇树网络。

算法设定：

(1) 当节点得票数超过参与阈值(joinThreshold)后才有权利参与对簇首的竞争。

(2) 节点保存的簇首状态数据结构myCH包含内容如表 2。

表 2 myCH数据结构

变量	意义
Id	本轮中节点的簇首节点
vote	簇首得票数
energy	簇首的能量
hoptosink	簇首到sink的跳数
hopdiff	本节点距离簇首跳数
Clusterparent	进行簇内路由时的父节点
Cparentenergy	父节点的能量

(3) 成簇请求报文(Cluster request message)格式如图3所示，各项意义与myCH结构各项意义相同。每一项长度为一个字节。

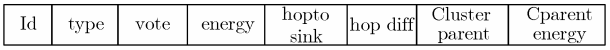


图 3 成簇请求报文格式

簇首竞争过程使用伪代码描述如下：

```
Begin
If (myvote>= joinThreshold)
{
    使用本节点状态设置myCH相应项;
    myCH→hopdiff=0;  //初始扩散跳数为0。
    myCH→Clusterparent=myid;
    myCH→Cparentenergy=myenergy;
    使用myCH初始化成簇请求报文p;
    broadcast(p);
}
End
```

节点接收到成簇请求：

```
Begin
If (节点监测值 > inspectThreshold){
    If (本节点已经将该请求中节点设置为簇首){
        If (收到请求中的hopdiff较小 || hopdiff相等但新收到请求的Cparentenergy较大)
        使用该成簇请求内容更新本节点myCH结构，以建立新的簇内路由;
        p→ hopdiff ++; //请求的扩散跳数加1。
        p→ Clusterparent＝myid;
        p→ Cparentenergy＝myenergy; //能量设为本节点能量。
        broadcast(p); }
        Else
        drop(p);
```

```
        }
    Else {//第一次收到该节点发出的成簇请求
        If (myCH中得票数<收到请求中的得票数 || 得票数相等但是请求中节点能量较大){
        使用该成簇请求内容设置本节点myCH结构;
myCH→Clusterpartent= p→ Clusterpartent;
//将父节点设置为这个请求的上一跳节点，作为簇内路由。
        p→ hopdiff ++;
        p→ Clusterparent=myid;
        p→ Cparentenergy=myenergy;
        broadcast(p); }
      Else
      drop(p); }
Else
drop(p);
End
```

如图4，采用Matlab对UVDC算法拓扑结构进行了模拟。25个节点均匀分布在50m×60m的区域内，阴影部分为事件区域。当成簇过程结束后，事件区域内得票最多的节点被选作簇首CH，并建立了以该节点为根的簇树网络。监测数据沿着簇内路径汇聚到簇首节点，在汇聚过程中中间节点对数据进行有限的数据融合，最后由簇首节点将所有数据融合在一起，将数据沿着网络主干传回sink节点。由于网络主干的构造不是本文关心的主要问题，因此在后面的仿真过程中采用了简单的最短路径算法来构造网络主干路由。

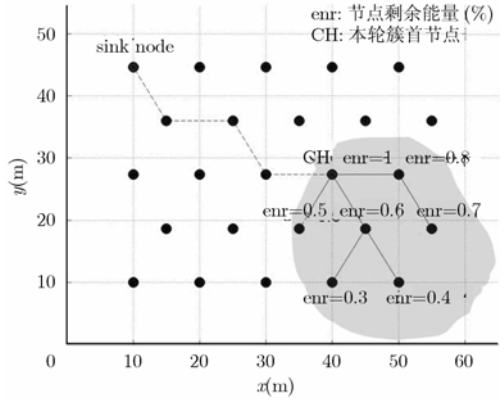


图4　UVDC算法网络拓扑

3.2　算法实现过程中两个细节问题

3.2.1 选举过程中的冲突问题　在选举过程中，由于事件的产生可能引起事件区域内的节点同时开始投票过程，而引起节点之间的冲突。我们通过以下两点解决可能出现的冲突问题：

(1) 算法中节点在发现监测值超过本身阈值的时候，并不立即投票，而是等待一个随机时间后将选票报文发送出去。这个退避时间的长短与本身监测值相关联，退避时间关系如下式：

$$T_{\text{backoff}} = T_{\min} + \left(\frac{\text{阈值}}{\text{监测值}}\right) \times (T_{\max} - T_{\min}), \quad \text{阈值} \le \text{监测值} \tag{3}$$

其中 $T_{\min}$ 为最小退避时间，$T_{\max}$ 为最大退避时间。监测值超过阈值愈多，退避时间越小。当监测值接近阈值时，$T_{\text{backoff}} = T_{\max}$；当监测值>>阈值时，$T_{\text{backoff}} = T_{\min}$。

(2) 选票报文是类似于 AODV 协议中 hello 报文的很短的帧，在采用退避算法后降低了发生冲突的可能性。

3.2.2 成簇过程中的广播风暴问题　广播风暴(broadcasting storm)是自组织网络中普遍存在的问题[9]。如在TEEN和LEACH的成簇过程中都需要作全网的广播来完成成簇任务，造成能量的消耗。在UVDC中通过以下 3 点来抑制成簇过程中的广播风暴：

(1) 选票报文只在局部有效，传播距离只有一跳，不被邻居节点转发。不会造成大范围的广播风暴。

(2) 只有具有选举权的节点才接收选票报文，否则将这些报文丢弃，将广播限制在事件区域内。

(3) 在节点竞争簇首过程中，节点会丢弃那些得票相对较少的成簇请求，不再广播出去。得票多的请求对得票少的请求产生抑制，减弱了广播风暴。

4　仿真结果分析与讨论

本文使用 NS2 对联合选举动态成簇路由算法进行了仿真分析。由于文献[1,2]所提出的动态成簇算法未深入考虑能量，因此本文未与其进行比较而是与成簇算法 TEEN 在节能方面进行了对比分析。仿真中使用一个通信半径为正常节点 1.5 倍特殊节点定期发出广播包来模拟一个事件，收到这钟包的节点认为监测到事件，然后生成一个大于监测阈值的随机值作为监测值。仿真中 25 个节点均匀分布在 50m×50m 区域内，inspectThreshold=5，joinThreshold =3，$T_{\max}$ =2s，$T_{\min}$ =1s。各节点初始能量为 20 ± 1J 内的随机值。UVDC 和 TEEN 的协议周期(round)都为 20s。TEEN 中成簇概率为 10% 。

4.1 平均能耗分析

图 5 对 UVDC 与 TEEN 的平均节点能耗进行了比较。从图中可以看出在不同仿真时间下，UVDC 的能耗都要优于 TEEN。主要由于成簇算法 TEEN 与监测事件的无关联性，导致事件区域不能与 TEEN 的预先成簇区域完全吻合，导致 TEEN 的数据融合效率不高，较多的元数据被发回到 sink 节点，造成了中间转发节点较多的能耗。而 UVDC 中事件区域的节点都在同一个簇内，具有较高的融合效率；其次 TEEN 中无论是否有事件发生都周期性在全网范围内重新成簇，增加了所有节点的平均能耗。UVDC 中每个协议周期只是事件区域内的少量节点参与成簇，在网络空闲时，UVDC 不必执行成簇工作。这种差距在网络负载较轻的情况下将更加明显。但由于 UVDC 为动态成簇，因此并不适合高负载的应用。

4.2 能耗均衡分析

对 UVDC 算法与 LEACH、TEEN 中簇首选举算法的能量均衡效果进行了仿真对比。在固定的事件区域分别采用了两种簇首选举算法。图 6 中给出了不同时刻事件区域各节点(a–g)的剩余能量曲线。看出使用 UVDC 算法的各节点能量曲线相对平坦。使用统计分析软件 SPSS 计算所得数据的相对标准偏差(RSD)。RSD 越小，说明各节点剩余能量越接近，即能耗越均衡。从表 3 可以看出 5 轮后，使用 UVDC 算法的网络节点剩余能量相对标准偏差仍维持在一个相对较低的水平，增长比较缓慢。根据相对标准偏差的意义，说明 5 轮后事件区域内节点 a–g 的剩余能量偏离不大，因为 UVDC 算法在选择簇首时，考虑了节点的剩余能量因素，当某个节点能量高出周围节点一定程度后，其将被选择为簇首节点，完成较多的数据搜集、融合、传输任务。而当能量降低后，则失去簇首的作用，算法选择其他能量相对较高的节点担当簇首。这样就不会造成能量过大或者过小节点的存在。

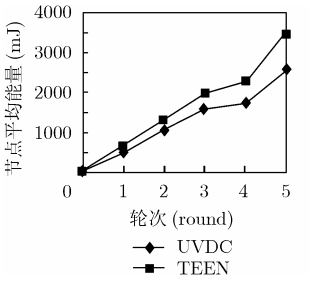


图 5 节点平均能耗

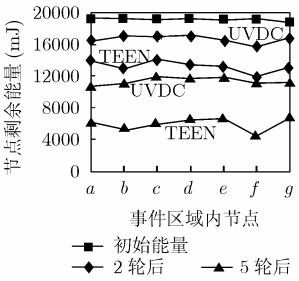


图 6 事件区域各节点能耗

表 3 节点剩余能量相对标准偏差

时间	相对标准偏差	
	TEEN	UVDC
初始时刻	0.01013	0.01013
2 轮后	0.04868	0.02575
5 轮后	0.13017	0.03861

在使用 TEEN 算法的网络中，节点剩余能量相对标准偏差变化明显，说明节点剩余能量偏离较大。主要原因为 TEEN 采用了“公平”的簇首选择机制，各个节点轮流担当簇首，而忽略了在各个节点担当簇首期间由于网络负载以及拓扑位置的不同而造成的能量消耗的差异。从而造成了节点剩余能量较大的偏离。

5 小结

本文针对无线传感器网络中突发事件监测应用提出了一种新的基于剩余能量的联合选举动态成簇算法。该算法考虑无线传感器网络中节能和能量均衡问题，利用簇首与簇成员能量消耗不平衡性实现事件区域内节点的能耗均衡。同时通过事件驱动动态成簇提高了数据融合的效率，减少了网络中传输的数据量，降低了节点的平均能耗。仿真结果表明，UVDC 能够降低节点的平均能耗，有良好的能耗均衡效果，延长了整个网络的生存时间，特别适用于无线传感器网络突发事件监测等的应用。

参 考 文 献

[1] Akyildiz I F, Su W, Sankarasubramaniam Y, and Cayirci E. Wireless sensor networks: A survey, computer networks. *The International Journal of Computer and Telecommunications Networking*, 2002, 38(4): 393–422.

[2] Heinzelman W R, Chandrakasan A, and Balakrishnan H. Energy-efficient communication protocol for wireless micro-sensor networks. The Hawaii International Conference on System Sciences, Maui, Hawaii, 2000: 3005–3014.

[3] Manjeshwar A and Agrawal D P. TEEN: A protocol for enhanced efficiency in wireless sensor networks. The 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, 2001: 189–196.

[4] Manjeshwar A and Agrawal D P. APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. The 2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile computing, Ft. Lauderdale, FL, 2002: 195–202.

[5] Chen W P, Hou J C, and Sha L. Dynamic clustering for acoustic target tracking in wireless sensor networks. network protocols. 2003 Proceedings. 11th IEEE International Conference on 4–7 Nov., 2003: 284–294.

[6] Fang Q, Zhao F, and Guibas L. Lightweight sensing and communication protocols for target enumeration and aggregation. Proceeding of the 4th ACM International Symposium on Mobile Ad hoc Networking & Computing, Annapolis, Maryland, USA June, 2003: 165–176.

[7] Jing A, Damla Turgut and Ladislau Bölöni. A cluster-based energy balancing scheme in heterogeneous wireless sensor networks. Networking-ICN 2005: 4th International Conference on Networking, Reunion Island, France, April, 2005: 17–21.

[8] Guerin R, Orda A, and Williams D. QoS routing mechanisms and OSPF extensions. The 2nd IEEE Global Internet Mini-Conference, Phoenix, AZ, Nov, 1997: 1903–1908.

[9] Sze-Yao Ni, Yu-Chee T, Yuh-Shyan C, and Jang-Ping S. The broadcast storm problem in a mobile Ad hoc network. Proceedings of the 5th annual ACM/IEEE International Conference on Mobile computing and Networking, Washington, United States August, 1999: 151–162.

郭 彬： 男，1980年生，博士生，研究方向为无线传感器网络、移动通信.

李 喆： 女，1967年生，教授，博士生导师，研究方向为无线传感器网络、Ad hoc网络、移动通信.