

密集型 WSN 冲突避免的 MAC 协议

史文博, 王继春, 李 响

(中国科学技术大学计算机科学与技术系, 合肥 230027)

摘 要: 无线传感器网络媒体访问控制层存在 2 种冲突。提出基于概率的时隙选择算法, 使不同节点在相同时隙发送概率不同, 从而降低域内冲突。实验结果显示, 该算法的有效能量和损失能量相比 Sift 协议节省了 17.6% 和 43.9%, 能量有效率提高了 14.3%。提出染色预防算法, 通过提前确定节点活动时序解决域间冲突问题。实验结果显示, 该算法的节点平均睡眠时间在 87% 以上, 空闲侦听导致的能量消耗仅占总能量的 7%。

关键词: 媒体访问控制协议; 基于概率的时隙选择; 染色预防; 冲突避免; 无线传感器网络

MAC Protocol with Collision Avoidance for Dense WSN

SHI Wen-bo, WANG Ji-chun, LI Xiang

(Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027)

【Abstract】 There are two kinds of collision in Medium Access Control(MAC) layer for dense Wireless Sensor Network(WSN). This paper proposes Probability-Based Slot Choice(PBSC) algorithm to solve intra-area collision, which chooses a transmitting slot based on probability. Experimental result shows that the algorithm attains a superior performance in effective energy and total energy by 17.6% and 43.9% compared with the Sift protocol. Coloring Precaution(CP) algorithm is proposed to solve inter-area collision, and the efficiency is improved by 14.3%. Because of low collision rate, average sleeping time is over 87% and energy for idle listening is below 7%.

【Key words】 Medium Access Control(MAC) protocol; Probability-Based Slot Choice(PBSC); Coloring Precaution(CP); collision avoidance; Wireless Sensor Network(WSN)

1 无线传感器网络媒体访问控制层传输冲突问题

无线传感器网络(Wireless Sensor Network, WSN)目前广泛应用于各种监测报警系统中, 图 1 为病房看护系统节点部署示意图, 传感节点通常放置在房间的某个位置(从节约成本出发, 允许出现一个节点同时负责 2 个病房的情况)。由于空间小、节点密度大, 不同节点可能同时感受到出现的异常情况(比如病人昏倒), 若它们同时发出报告, 无线信道就产生了冲突。此外, 不同病房的 2 个病人可能同时出现紧急情况, 那么负责监视 2 个相邻房间的传感器就会同时感受到 2 条不同的信息, 如果它对信息 I_i 做出响应, 势必会影响监测到信息 I_j 的其他节点及时汇报。

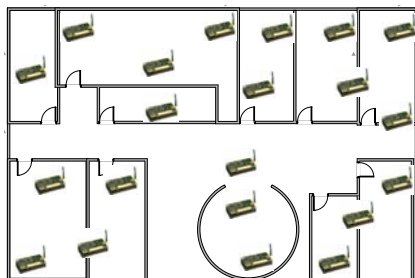


图 1 病房看护系统节点部署示意图

不失一般性, 本文把问题抽象地描述成: 网络包含 N 个传感节点 $S_i \in S, i=1,2,\dots,N$ 及 M 个信息发生地点 $p_j \in P, j=1,2,\dots,M$, 这 N 个节点和 M 个信息产生地点随机地分布在被监视区域的各个位置, 每条报警信息 I_i 由一个三元组唯一标识:

$$I_i = \langle X_i, Y_i, TS_i \rangle \quad (1)$$

其中, X_i, Y_i 分别表示异常情况发生地点 p_i 的二维空间坐标; TS_i 表示信息产生的时间。节点 S_j 的表示形式是

$$S_j = \langle X_j, Y_j \rangle \quad (2)$$

其中, X_j 和 Y_j 是节点 S_j 的空间位置。对于这样的网络, 异常情况可能被 m 个节点同时感受到, 所以, 感知信息会在一瞬间迅速膨胀。本文的目标是确保 m 个节点形成的 m 条感知信息至少有一条以最快速度到达基站。因此, 密集型网络的媒体访问控制(Medium Access Control, MAC)协议就是保证 MAC 层能把至少一条关于异常信息 $I_i = \langle X_i, Y_i, TS_i \rangle$ 的拷贝无冲突且快速可靠地传递到基站。

2 密集型 WSN 模型

本文研究的无线传感器网络满足以下 5 点假设:

(1) 无线信道分成控制信道和数据信道^[1], 控制信道仅用于唤醒节点, 大量信息的发送只能依靠唯一的数据信道。不同于 Single Radio Multi-Channel^[2] 或者 MIMO(Multi-Input Multi-Output)^[3] 模式, 节点对信道的占用必须交替进行, 否则发生冲突。

(2) 所有节点的感知、通信、计算、存储能力完全相同, 初始能量配置也完全一样, 没有普通节点和超级节点之分。

(3) 节点的感知范围和通信范围都是规则的圆形, 其中,

作者简介: 史文博(1983-), 女, 硕士研究生, 主研方向: 无线传感器网络 MAC 协议和拓扑控制; 王继春, 博士研究生; 李 响, 硕士研究生

收稿日期: 2009-06-30 **E-mail:** shiwenbo@mail.ustc.edu.cn

感知半径小于通信半径。

(4)节点一旦部署完毕,其物理位置不再发生改变。它们主要执行2种任务:监测异常情况和传递数据。

(5)节点为每个异常情况独立生成报告信息,当感知到2个或2个以上的异常情况发生时,分别形成多条不同的报告信息,数据转发时可以一次连续传递寄存器中的全部数据。

3 MAC协议冲突类型

密集型网络瞬间信息膨胀将导致冲突问题。 M 个信息源每一时刻都按一定的概率随机产生异常情况,节点 S_i 与信息源 p_j 的欧式距离 $d(S_i, p_j)$ 为

$$d(S_i, p_j) = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \quad (3)$$

一旦出现异常情况,与该出事地点距离小于感知半径的节点 S_i 就产生报警信息,所以,节点产生报警信息的条件是

$$d(S_i, p_j) \leq Sen_Range \quad (4)$$

其中, Sen_Range 是感知半径。对报警状态的节点进行划分,感知到相同报警信息的节点组成的集合叫作一个信息域 A_k ,定义为

$$A_k = \{S_i | S_i \text{ sensing } I_k\} \quad (5)$$

其中, $I_k = \langle X_k, Y_k, TS_k \rangle$ 。由于报警信息需要以最快速度传到监控中心(基站),因此一个信息域的节点同时报告信息就可能发生信道使用时的碰撞,这时发生的冲突定义为域内冲突,它产生的条件是满足:

$$(S_i \in A_k) \cap (S_j \in A_k) \quad (6)$$

为了便于讨论问题,给出以下相关定义:

(1) A_k 的邻居集 $N(A_k)$,表示信息域 A_k 中节点的所有一跳邻居的集合。

$$N(A_k) = \{S_j | (S_i \in A_k) \cap (d(S_i, S_j) \leq Con_Range)\} \quad (7)$$

其中, Con_Range 表示通信半径。可见 $N(A_k)$ 反映了 A_k 的影响范围。

(2)域间冲突,与域内冲突相对应,产生的原因是不同报警信息的2个信息域过于邻近甚至交叉,导致2个信息域 A_k 与 A'_k 的公共元素发生选择分歧。域间冲突产生的条件是符合

$$(A_k \cup N(A_k)) \cap (A'_k \cup N(A'_k)) \neq \emptyset \quad (8)$$

(3) A_k 的冲突域 $C(A_k)$,表示可能发生域间冲突或域内冲突的所有点的集合。

$$C(A_k) = A_k \cup N(A_k) \quad (9)$$

根据上述定义,图2给出一个产生冲突的例子,共产生报警信息 $I_1 \sim I_8$,所以,形成8个信息域 $A_1 \sim A_8$ 。

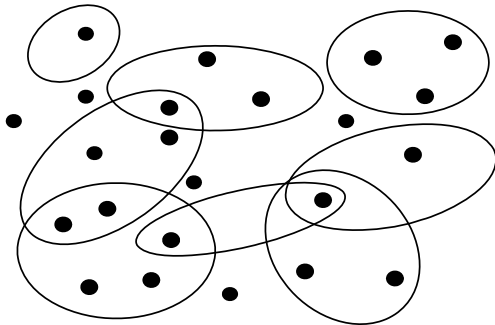


图2 网络信息域的划分

本文的目标有2个:(1)降低域内冲突的可能性,即通过有效的时隙选择方法,同时兼顾节点个体能量的消耗,防止

同一个信息域内的不同节点同时选中同一个时隙。(2)及早预防可能发生的域间冲突。

4 域内信息报告

由于一个信息域内分布密集的节点同时监测到一个异常信息时,这些节点对该信息的报警具有相同的义务,它们的反应时间也大致相同,因此如果采用纯粹的基于竞争的CSMA方式占用信道,势必会引发大量冲突,利用可变窗口随机退避的算法^[4]会导致等待时间随信息域内节点数目的增多而迅速增长,从而违背快速传递的要求。文献[5]提出的Sift协议采用了概率选择的办法,比较适合这种应用背景。但是Sift协议的缺陷是全体节点执行单一的概率函数,导致节点或者都不发送,或者许多节点选择同一个时隙同时发送,仍存在较高的风险发生冲突。

4.1 域内冲突解决策略

为了弥补可变竞争窗口的退避算法及Sift协议的不足,本文基于概率选择的方法进一步使节点的选择时隙“个性化”,即让不同的节点执行不同的函数进行概率选择,把发生冲突的风险进一步“离散化”,使每个节点在不同的时隙具有不同的概率分布,原理如图3所示。

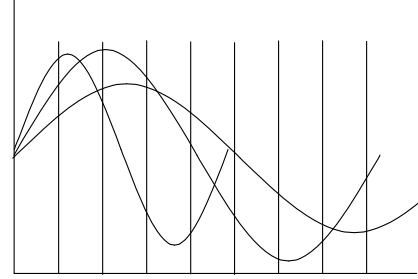


图3 节点按概率快速选择slot的原理

域内节点在感知到异常信息后,会经过一个短暂的“思考”时间,然后决定报告出去。如果每个节点被允许的最长思考时间相等,把它划分成更小的时间单位slot, T 表示最大slot数目,则节点相当于在 $[0, T]$ 中选中某个slot发送数据。节点彼此越相似,它们选择同一个slot的概率就越大。本文充分发掘了节点的多样性,让它们以非均匀的方式选择slot,从而破除这种相同的偏好,减小冲突的可能性。换言之,要突出节点的差异性,找到每个节点区别于其他节点的特性,使节点根据这些特性对每个slot以不同的概率进行选择,从而降低与其他节点同时选择同一个slot的几率。

对实时系统的信道占用起决定作用的往往是最前面的几个时隙,因为一旦有一个竞争者胜出,本次竞争就宣告结束。之后胜出的节点迅速向其下一跳汇报此报警信息,以减少信息在传递中的耽搁。

4.2 基于概率的时隙选择算法

节点分3步完成时隙的选择:(1)通过线性加权的办法得到自身的Rank。(2)根据Rank的大小确定自己在每个slot发送的概率。(3)slot来临时,依据概率在发送和不发送这2个动作间选择其一。

一个信息域内的节点根据Rank的不同,在第 t 时隙,有的以高概率选择slot,有的以低概率选择slot,但是,并非高概率的节点就一定会发送,相反,以低概率选择的节点完全有可能竞争到信道。这种用概率选择的方法只是为了突出各节点的个体差异,降低它们同时传输的几率,因此,需要尽

量让节点快速做出决定，同时不受其他节点的干扰。

下面研究节点如何得到自身的 Rank。节点独有的特征包括剩余能量、距离基站的跳数、地理位置是否处于更重要的交叉区域等，每个方面都是影响全局的一部分。所以，可以用向量的形式表示这些线性无关的影响因素。

$$\mathbf{R} = [r_1, r_2, \dots, r_m] \quad (10)$$

其中， r_1, r_2, \dots, r_m 分别表示 m 个参考因素。

由于 m 个方面所起的决定作用不一定相同，因此为每个分量分配一个权值。在 WSN 中能量是最受关注的指标，因此，能量因素的权值相应地大于其他因素的权值。 \mathbf{R} 对应的权值向量表示成 $\mathbf{W} = [w_1, w_2, \dots, w_m]^T$ ，其中，

$$\sum_{i=1}^m w_i = 1, w_i \in [0, 1] \quad (11)$$

节点 S_i 由此得到它的综合 Rank:

$$\text{Rank}(i) = f(r_1, r_2, \dots, r_m) = \mathbf{R} \cdot \mathbf{W} = \sum_{i=1}^m r_i \cdot w_i \quad (12)$$

Rank 的高低决定了此节点在 slot 选择上具有的不同概率。本文旨在让节点自己为每个 slot 分配一个概率，即节点在每个 slot 都有机会发送，只是概率不同，有的节点在该 slot 发送的概率高些，有些节点发送的概率低些，由此降低节点同时选择一个 slot 的几率。

综合考虑节点自身状况的另一个好处是可以平衡节点之间的能量消耗，延长整个网络的生命周期。既然 WSN 最关注的是能量问题，那么可以分配给能量因素更高的权值，使剩余能量高的节点优先占用前几个 slot。

基于概率的时隙选择 (Probability-Based Slot Choice, PBSC) 算法过程如下:

```

if ( $S_i$  detects an Abnormity  $I_i$ ) //节点  $S_i$  感知到异常情况
S1: then  $S_i$  gets Rank(i); //节点结合能量计算自身的 Rank(i)
for each slot  $L_j$  in  $[0, T]$  calculates probability  $P_j$  according to Rank(j); //计算自己在  $[0, T]$  每个时隙的发送概率
if ( $L_j$  is coming) //时隙  $L_j$  到来时
then  $t = \text{random}()$ ; //随机产生一个  $[0, 1]$  之间的数  $t$ 
if ( $t \leq P_j$ )
then Send_data(); //报告消息
else wait next slot  $L_{j+1}$ ; //等待下一个时隙
if (slot  $\leq T$  and no node  $S_j$  decides to send) //未到最大时隙且
//无其他节点发出报警
then goto S1;
else 此轮时隙选择结束

```

5 域间冲突的消除

网络中产生的各种异常信息可以看作按照产生的先后顺序依次进入一个信息队列，该队列遵从先进先处理的原则，一旦信息到达基站，就将它从队列清除，否则，表示此信息正在网络中被传递。如果信息域交叉，将会引起域间冲突。

5.1 域间冲突

在实际应用中，节点一旦布置就基本不再移动，因此，根据位置坐标可以判断，只要出事地点在某个节点的感知范围内，一旦发生异常情况，节点就一定会发出警报。基于以上分析，网络部署完后，先要进行冲突检查。假设一个信息域 A_i 与 A_j 中任意节点相距一跳的邻居节点组成集合 $N(A_i)$ ，冲突产生条件为 $(A_i \cup N(A_i)) \cap (A_j \cup N(A_j)) \neq \emptyset$ 。如果不采取任何措施，一旦网络开始运行，信息域 A_i 中的节点抢着报告信息 I_i ，信息域 A_j 中的节点忙着报告信息 I_j 。假设 2 个冲突域

存在公共元素 S_k ，则某个时刻在 S_k 处，2 个信息 I_i 和 I_j 发生碰撞，导致 2 个信息都被损坏。为避免这种情况的发生，必须使 I_i 和 I_j 在到达 S_k 的时间上做到有先有后。因此，采用图的染色办法，使相互关联的区域在时间上错开，只要存在一种被普遍认可的先来后到的时序关系，那么 2 个信息域发生关联时，就会按照这种约定俗成的位次关系顺序执行，避免了信息碰撞的情况发生。这是一种用时间换空间的方法，由于资源的限制，需要在某些特殊地点把空间上多维并行的几个操作就地排成一维线性的列表，按照不同的时间段依次执行以解决冲突。但是为了满足及时传递的需要，应该尽可能使用更少的时间段，使消息传递的并行处理程度最大化，这样既解决了冲突也不会造成过大的时延。

5.2 染色预防算法

本文根据拓扑图构造冲突图 $G = (V, E)$ ，构造原则是：原拓扑图中由众多点组成的一个信息域 A_k 在冲突图中被浓缩成一个点，用 V_k 表示。如果原图中有 M 个信息域，新图中就对应 M 个点，所以，冲突图的点集合表示成

$$V(G) = [V_1, V_2, \dots, V_M] \quad (13)$$

如果 2 个冲突域 $C(A_i) = A_i \cup N(A_i)$ 与 $C(A_j) = A_j \cup N(A_j)$ 有公共节点，则在 G 中， V_i 和 V_j 之间连接一条边 E_{ij} 。于是冲突图的边集合表示成

$$E(G) = \{E_{ij} \mid C(A_i) \cap C(A_j) \neq \emptyset\} \quad (14)$$

本文的思想是先对冲突图中 M 个点按照它们的度由大到小降序排列 $\delta(V_{i1}) \geq \delta(V_{i2}) \geq \dots \geq \delta(V_{iM})$ ， $\delta(V_k)$ 越大，表示冲突域 $A_k \cup N(A_k)$ 中包含的节点个数越多，它对周围的影响力就越大。所以，每次都优先选择度最大的尚未分配时间段的节点进行染色。新图中一旦一个点 V_k 被染上第 i 种颜色，就意味着原图中这个信息域 A_k 中的所有节点都被分配到了第 i 个时间段。

染色预防 (Coloring Precaution, CP) 算法过程如下:

阶段 1 前期准备

Step1 基站广播数据包，宣布自己是树的第 0 层。

Step2 周围节点根据与基站的距离把自己定位在第 1 层，把基站作为父节点，继续广播。

Step3 收到广播包的节点把其中 (不止一个广播包) 的最小层数增加 1 作为自己在树中的位置，同时把该广播包的发出者设置为自己的父节点。

Step4 只要尚有节点未加入，就返回 Step3。

Step5 所有节点找到父节点，前期准备结束。

阶段 2 染色划分时间段

Step1 初始时， $color=0, V[color]=\emptyset$ 。

Step2 考察所有信息域，选出冲突域 $C(A_i)$ 包含节点数目最多的信息域 A_i ，为 A_i 的所有节点分配第 1 个 $color$: $V[color]=\{A_i\}$ 。

Step3 查看剩余信息域，找出冲突域 $C(A_j)$ 包含节点数最多的一个信息域 A_j 。如果 A_i 与 A_j 的冲突域不相交，则将它们分配在相同 $color$ ，并扩大该 $color$ 包含的信息域集合；否则，增加一个 $color$: $color=color+1$ ，因为存在冲突危险。

Step4 只要尚有信息域未被分配 $color$ ，返回 Step3。

Step5 $color=color+1$ 。因为其余不属于任何信息域的大量节点不存在任何制约关系，所以为它们另外分配一个 $color$ 。

阶段 1 的工作是组成一棵以基站为根节点、包含全部节点的树型结构，作为信息传递的路径；阶段 2 的染色过程根据信息域 A_i 与 A_j 的相关程度利用 Greedy 策略求得一种分配方案，如图 4 所示。这种方案使分属于不同信息域的节点要么可以在一个时间段内并行处理，要么在 2 个时间段内分别处理，由此消除域间冲突。

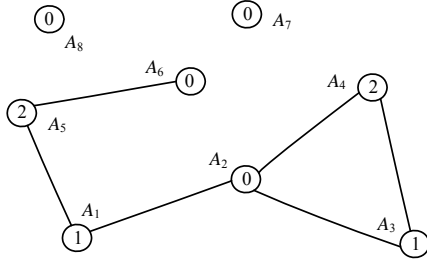


图 4 冲突图经过染色时间段分配方案

6 实验模拟与仿真

在 $200\text{m} \times 200\text{m}$ 的区域中，随机布置 100 个节点、10 个异常信息源，并随机选定一个节点作为基站。节点的传感距离 $SR=25\text{m}$ ，通信距离 $CR=35\text{m}$ 。在 MFC 仿真环境中，分别从 3 个方面分析算法的性能。实验参数的设定如下：信道带宽为 2Mb/s ；发送能量为 24.75mW ；接收能量为 13.5mW ；空闲能量为 13.5mW ；睡眠能量为 $15\mu\text{W}$ ；回退窗口大小为 32 Byte；ACK 帧长为 112 bit；RTS 帧长为 160 bit；CTS 帧长为 112 bit；有效荷载为 8184 bit；ACK/CTS 超时为 $300\mu\text{s}$ ；侦听时间为 $0\mu\text{s}$ 。

6.1 空闲比性能评价

在本算法中，节点不需要执行严格的固定调度。发送/接收时间是节点完成任务必不可少的状态。只有空闲侦听阶段属于能量的无谓消耗，因此，本文用 ρ' 评价算法的性能：

$$\rho' = \frac{T_{\text{idle}}}{T_{\text{sleep}} + T_{\text{tran}} + T_{\text{rev}} + T_{\text{idle}}} \quad (15)$$

在节点的生命周期里，发送状态时间 T_{tran} 和接收状态时间 T_{rev} 都是由网络负载决定的，而空闲侦听时间 T_{idle} 的长短才真正体现节点能量的消耗情况。 T_{idle} 所占的比例越小，表明节点浪费的能量越少。图 5 是在不同拓扑条件下节点处在各个状态时间比例的比较。

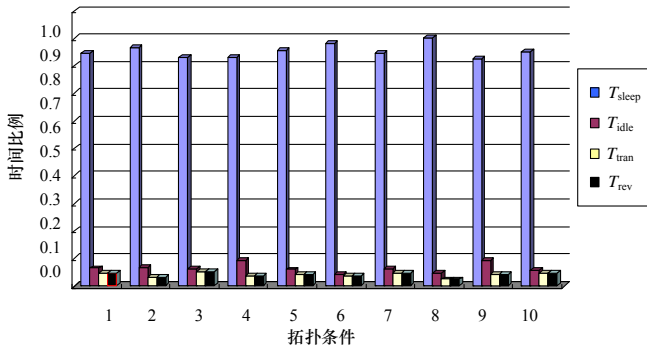


图 5 节点处于各拓扑条件下的时间对比

针对不同的拓扑条件，计算各状态平均消耗的时间比例，可得节点在生命周期中，87%的时间处于睡眠状态，而 ρ' 空闲侦听的时间占全部时间的 7% 左右。

6.2 能量消耗

由于本文算法与 Sift 协议都是节点利用概率对发送时隙

进行选择，因此可以在节点数目相同的情况下比较 2 种协议的能耗。

设节点个数为 200 个，窗口数为 12 个，异常信息发生率是 0.5，感知到异常信息的节点数目在 50~150 之间且均匀分布，Sift 协议与 PBSC 协议能量消耗如图 6 所示。完成相同条件下的信息报告，PBSC 协议的有效能量发送为 Sift 协议的 82.4%，对于出现冲突后损失的能量，PBSC 协议占 Sift 协议的 57.1%，因此，总能量节约了 26%。

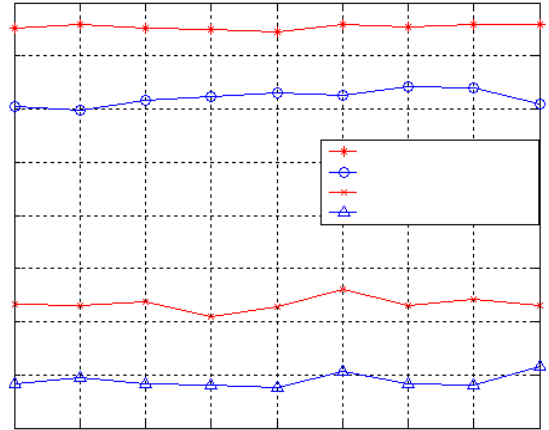


图 6 Sift 与 PBSC 能量消耗的比较

为了进一步分析 PBSC 的性能，本文给出能量有效率的定义：

$$\eta = \frac{E_t}{E_t + E_c} \times 100\% \quad (16)$$

其中， E_t 表示成功传输的能量； E_c 表示遇到冲突损失的能量。根据式(16)得到 Sift 协议和 PBSC 协议的能量有效率，如图 7 所示。从中可以看出，PBSC 协议的能量有效率提高了 14.3%。

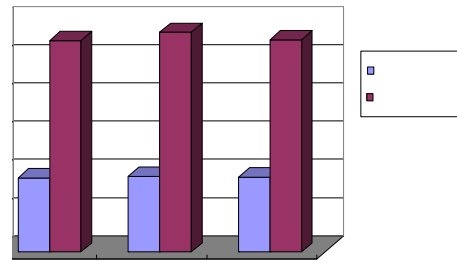


图 7 Sift 与 PBSC 能量有效率的比较

7 结束语

本文描述了密集型无线传感器网络 MAC 层遇到的问题，根据产生的原因将冲突划分为域内冲突和域间冲突。针对域内冲突，提出基于概率的时隙选择算法，即参考节点自身能量、位置等指标，加权求得节点的综合排名，使节点按不同概率选择发送时隙，从而降低了它们同时发送的概率。为避免域间冲突，提出了染色预防算法，使潜在的冲突危险被提前化解。实验结果表明，在频繁发生异常信息的网络中，算法因为有效降低了冲突，避免了能量大量浪费在空闲侦听上，所以保证了节点平均睡眠时间在 87% 以上。而局部区域中基于概率的时隙选择算法比 Sift 协议在有效能量和损失能量上分别节省了 17.6% 和 43.9%，能量有效率提高了 14.3%。

(下转第 105 页)