软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(6):1316−1327 [doi: 10.13328/j.cnki.jos.004437]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于冲突指示和分组隐藏节点冲突解析策略*

李拥军[1], 谢 嵘[1], 谭晓青[2]

[1](华南理工大学 计算机科学与工程学院,广东 广州 510006)

[2](暨南大学 信息科学技术学院,广东 广州 510632)

通讯作者: 谢嵘, E-mail: rong_xie@126.com

摘 要: 隐藏节点问题是导致 IEEE 802.15.4 协议性能下降的一个重要因素,而在 IEEE 802.15.4 中没有给出解决该类问题的具体方案.提出一种基于冲突指示和分组的隐藏冲突避免策略(hidden node collision detection and avoid strategy,简称 HNCDAS),该策略采用分组方法将 IEEE 802.15.4 的 CAP 周期划分为多个等分时隙,从隐藏冲突导致的部分破损帧中提取出隐藏节点地址信息,依据当前获得的隐藏关系动态地将节点调整到相应的竞争组,竞争组内的节点在同一周期内仍按照二进制后退方法竞争发送消息,不同的竞争组在不同的时隙发送消息,从而彻底解决隐藏冲突问题.与其他隐藏冲突解析策略相比,HNCDAS 具有额外开销少和动态调整等优点.从理论上证明了该策略的收敛性和解析策略时间的上限,实验结果表明,HNCDAS 在数据传递率、吞吐率和能量利用率等方面都有明显的提高.

关键词: IEEE 802.15.4;隐藏冲突问题;隐藏冲突解析机制;分组策略;LR-WPAN

中图法分类号: TP393

中文引用格式: 李拥军,谢嵘,谭晓青.基于冲突指示和分组隐藏节点冲突解析策略.软件学报,2014,25(6):1316−1327. http://www.jos.org.cn/1000-9825/4437.htm

英文引用格式: Li YJ, Xie R, Tan XQ. Resolution Strategy of Hidden Node Collision Based on Collision Indication and Grouping. Ruan Jian Xue Bao/Journal of Software, 2014,25(6):1316−1327 (in Chinese). http://www.jos.org.cn/1000- 9825/4437.htm

Resolution Strategy of Hidden Node Collision Based on Collision Indication and Grouping

LI Yong-Jun[1], XIE Rong[1], TAN Xiao-Qing[2]

[1](School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China)

[2](College of Information Science and Technology, Jinan University, Guangzhou 510632, China)

Corresponding author: XIE Rong, E-mail: rong_xie@126.com

Abstract: Hidden node problem is an important factor in performance degradation of IEEE 802.15.4 protocol. This paper presents a resolution strategy of hidden collision based on collision indication and grouping. The new strategy, named Hidden Node Collision Detection and Avoidance Strategy (HNCDAS), uses grouping method to divide the CFP of IEEE 802.15.4 protocol into several equal slot cycles and extract the hidden node address information from some damaged frame caused by the hidden conflict. The strategy dynamically adjusts nodes to different competition groups based on currently obtained hidden relationship. The nodes within competitive groups still competitively send messages in accordance with the binary back method in the same period, while different competitive groups send messages in different time slots. As a result, the strategy completely solves the hidden conflict problem. Compared with other hidden node collision resolution strategies, HNCDAS has certain advantages such as less overhead and dynamic adjustment capability. The convergence of the strategy and the maximum time of resolution strategies are also demonstrated in theory. Experimental results show that HNCDAS can significantly improve data transmission rate, throughput and energy efficiency.

* 基金项目: 国家自然科学基金(61003258, 61370228); 国家重点基础研究发展计划(973)(2007CB311100); 中国博士后科学基金(20110490884); 广东省科技项目(201079, 2011B010200039, 2012A010701006); 广州市科技项目(11C42080722)

收稿时间: 2012-08-22; 定稿时间: 2013-06-09

© 中国科学院软件研究所 http://www.jos.org.cn

Key words: IEEE 802.15.4; hidden node problem; collision resolution strategy; grouping strategy; LR-WPAN

无线传感器网络是一种新型的网络和计算技术,它具有自适应性、抗毁性、容易部署和低成本等优点,在环境监测、工矿企业信息收集、恶劣环境数据收集、军事与国防应用领域有着广泛的应用[1,2].IEEE 802.15.4 协议是一种低功耗、短距离、低速率、低复杂度的无线接入协议,是目前无线传感器网络中的主要 MAC 协议.

802.15.4 网络的节点数量大,分布随机性强.文献[3]说明,在节点随机分布的网络中,发生隐藏冲突的概率高达 41%,频繁的隐藏冲突导致大量的消息重传,不仅会降低网络的吞吐量、增加端到端通信延时,而且会导致大量无谓的能量消耗,极大地降低了整个网络的生存周期,节能是无线传感器网络首要考虑的问题[4].隐藏冲突所造成的网络性能下降和大量能源无谓消耗是 802.15.4 亟待完善的问题.

一般来说,无线网络中存在两种冲突:一种是竞争引起的冲突(contention collision),另一种是隐藏节点引起的冲突(hidden node collision,简称 HNC).竞争冲突可以通过二进制指数后退算法(TBEB)来解决;对于隐藏冲突,802.11 使用 RTS/CTS 握手协议[5,6]来消除,而 802.15.4 则没有采用任何保护机制来防范,其主要原因为:

(1) 在没有数据发送的情况下,802.15.4 节点的无线收发器是关闭的,接收不到 RTS 包;

(2) 由于 802.15.4 MAC 帧较小(最大为 127byte),发送 RTS/CTS 包产生冲突的可能性与传送数据帧一样,采用 RTS/CTS 机制并不能有效地抑制隐藏冲突,还会导致多余的能量消耗.因此,RTS/CTS 握手协议不适用于 802.15.4 网络.

除了用 RTS/CTS 机制来抑制隐藏冲突外,另一个解决隐藏冲突问题的方法是 busy tone[7−9],它们大多以 RTS/CTS 为基础,需要额外的频道传送 busy tone 信号,因此需要在硬件上增加额外的无线收发器.而文献[10,11]则在不增加额外频道的情况下使用 busy tone 的概念,但是网络中的节点为了接收 busy tone 信号,必须时常打开接收器,因此造成多余能量的消耗,不适应于无线传感器网络这种能量严格受限的应用.文献[12−14]则提出了利用控制发送功率的方法来减少隐藏冲突问题,但是提高发送功率会增加接收端受干扰的范围,同时也增加了节点的能量消耗.

文献[15,16]对无线网络提出了分组的概念,其中,文献[16]提出了一种分组策略(grouping strategy)解决隐藏节点问题,解决步骤分为 4 步:隐藏节点场景发现、隐藏关系收集、节点分组、带宽分配.算法的具体思路是:根据节点间的隐藏关系把节点划分为 N 个只有竞争冲突的组,然后将原来的自由竞争时间片(CAP)均匀分成 N 个时间片,组内节点只能在规定的时间片内竞争发送数据,但如果某一时间片所属的组没有节点要发送数据,其他组也不能使用该时间片.最后证明,对某一节点来说,与之构成隐藏关系的最大分组数是 5.但该文适用的范围是星型网络;同时,该分组策略的主要问题是算法的消息复杂度高为 $O(N^2)$,会消耗节点大量能量.尽管还有一些方法[6,17]来降低消息复杂度,但它们都不适应节点移动的场景.文献[18]则针对某些应用节点发送数据,提出了不均匀的时间片划分方法.

Sheu 等人提出了一种 P-frozen 策略[19],该策略假设协调节点能够从隐藏冲突破损的帧中提取出地址和长度信息,然后由协调节点给相应的节点分配 GTS 时间片,该节点将在专用的 GTS 时间片往协调节点发送数据.然而,P-frozen 策略中节点还需要有退出 GTS 的操作,如果在节点发送数据量不大的情况下,频繁地加入和退出 GTS,会导致能耗增加.

本文结合分组策略与 P-frozen 策略提出一种基于冲突指示和分组的隐藏节点冲突避免策略(hidden node collision detection and avoid strategy,简称 HNCDAS).第 1 节提出基于冲突指示的隐藏关系发现方法,在此基础上给出基于冲突指示和分组的隐藏冲突解析策略及示例.第 2 节对算法的收敛性和收敛时间上限进行理论证明.第 3 节将改进的算法与标准的 802.15.4 协议进行比较和分析.最后对本文进行了总结.

1 基于冲突识别和分组的隐藏节点冲突解析策略

1.1 基于冲突指示的隐藏关系发现方法

定义 1. 在无线网络中,如果节点 A 向节点 B 发送消息,而节点 C 没有侦听到,则 A 和 C 相互构成隐藏关系.

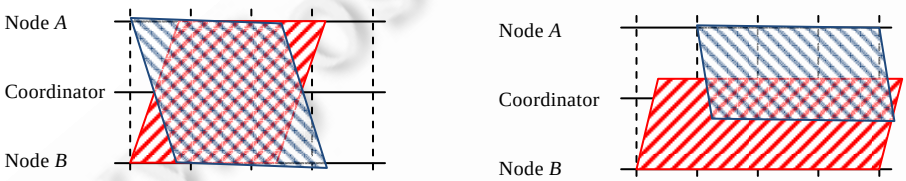
© 中国科学院软件研究所 http://www.jos.org.cn

发现节点间的隐藏关系是解决隐藏冲突的前提条件.目前,隐藏关系的发现算法主要有两种:

(1) 隐藏信息收集算法.节点在协调器节点的控制下,逐个向协调器节点发送数据包,其余节点处于监听状态,如果在规定时间内没有侦听到数据包,则与发送节点构成隐藏关系.最后,节点将收集到的部分隐藏关系发送回协调器,协调器节点根据汇总的信息,构建出一幅隐藏关系图.这种算法是一种集中式算法,消息复杂度高;

(2) 加入时确定隐藏关系,新节点在加入 cluster 前需要向协调器节点发送 join 消息,而 cluster 内的节点处于监听状态,如果接收不到,则认为与新节点构成隐藏关系.这种方法必须假设新节点加入前让 cluster 内所有节点处于监听状态,这一假设要求较高,在实际应用过程中,无线接收模块处于关闭状态,簇内节点往往处于休眠状态,因而这种方法也不适用于实际应用.

当多个节点向协调器节点发送数据时,数据包会在协调器端产生叠加,这种数据包重叠有两种情况:起始部分重叠(如图 1(a)所示)(两个隐藏/竞争节点选择同一时刻发送数据)和部分重叠(如图 1(b)所示)(在一个节点发送数据过程中,另一个隐藏节点发送数据).由于起始部分重叠只有在节点选择同一时刻发送数据才会发生,而部分重叠在整个数据发送过程中都会发生,所以部分重叠发生的概率要远远高于起始部分重叠.对于部分重叠的情况,在协调器接收到的部分叠加数据包中,有一部分信息未被损坏,我们可以保留起来,并加以利用.



(a) 起始部分重叠 (b) 部分重叠

Fig.1 Two cases overlapping data

图 1 两种数据重叠的情况

文献[19]提出的 P-Frozen 策略假设协调器节点底层硬件能从部分叠加而未破损的帧中提取出头部地址和帧长信息,然后由协调节点给相应的节点分配专用的 GTS 时间片,并通知该节点让其在专用的 GTS 时间片发送数据(如图 2 所示).我们对该方法加以改进,假设底层硬件具备如下能力:① 协调器节点在检测到破损帧时,继续接收数据包,直到帧尾;② 占用子节点的数据帧负载部分尾部 3 个字节,分别存放尾同步字段、源地址信息和 CRC.其中,尾同步字段用于当两个数据帧发生隐藏冲突时,协调器节点利用该同步头定位尾部附加字段的起始,而 CRC 则用以校验尾部的地址信息是否被破坏(如图 3 所示).当两个数据包发生隐藏冲突时,协调器可从接收到的部分破损数据包的头部和尾部分别提取出这两个节点的地址,这两个地址构成一对隐藏关系.从而使协调节点能够识别出到底是哪两个节点发送隐藏冲突,协调器节点保存该隐藏关系,并启动第 1.2 节的动态分组算法,根据隐藏关系,将节点划分到不同的分组,实现彻底消除隐藏节点的目的.虽然数据帧有效负载减少了 3 个字节,但可以利用该附加信息识别隐藏关系,利用分组策略避免隐藏冲突,获得更高的数据到达率和节能目的.

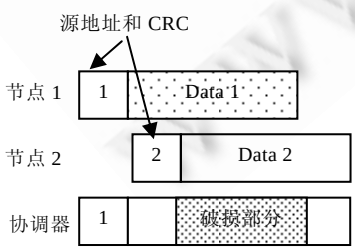


Fig.2 The Strategy of P-Frozen

图 2 P-Frozen 策略

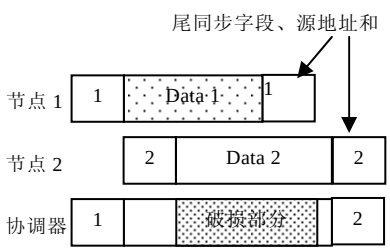


Fig.3 Additional Fields of HNCDAS

图 3 HNCDAS 策略尾部附加字段

© 中国科学院软件研究所 http://www.jos.org.cn

但是,在以下情况下不一定能正确提取出隐藏关系:① 如图3节点2发送的时间往前提前,则有可能把节点1的地址信息也给覆盖掉了;② 节点1的数据帧较长,覆盖了节点2的数据帧尾;③ 在网络中数据到达率较高时,可能节点1之后有节点3进行数据帧传输,与节点2帧也发生冲突,在鉴别其不为竞争冲突之后,以同样的方式确定节点3与节点2的隐藏关系.在前面这些情况下,如果提取不出隐藏关系,则不启动动态分组算法.

1.2 基于冲突指示和分组的隐藏节点冲突解析策略

基于分组的隐藏冲突解析策略主要由隐藏关系发现和分组两部分构成,其中:隐藏关系发现是指通过各种技术手段发现网络中存在隐藏冲突的节点,它是隐藏冲突解析的先决条件;而分组则是根据收集到的所有隐藏关系将网内节点划分为多个节点间不存在隐藏关系的组,并将超码框的CAP周期划分给相应的组使用,时间周期的划分方法有:均匀划分和非均匀划分.

本文的侧重点在于隐藏关系的发现算法,根据我们在第1.1节提出的方法,当发现一对隐藏关系后,开始启动动态分组算法HNCDAS,将这对隐藏关系中的节点划分到不同的竞争组,并将CAP周期划分为多个等分时隙,为每个分组分配一个时隙,具体算法如下:

```
Initial: G={{v1,v2,…,vn}}; H=∅;
//G 为当前已有的竞争组,H 记录已有的隐藏关系,{va,vb}为输入的一对隐藏关系
Adjust_group(G,H,{va,vb})
{   Check_move(G,H,va);  //检查与节点 va 构成隐藏关系的组,如果组大于 5 个,说明节点 va 发生了移动
    Check_move(G,H,vb);
    If {va,vb} in H then return;   //如果隐藏关系已存在,则返回
    H=H∪{{va,vb}};   //将 va 和 vb 加入隐藏关系列表
    //va,vb 属于同一个组 gt,由于 va 与 vb 相冲突,将 vb 从 gt 组取消,并将其划分到别的组.va 不进行调整是因为 va 节点数据发送完,等待协调器节点的 ack,如果没等到则休眠,而 vb 节点这时还在等待 ack,我们可以通过发一个 NAK,通知 vb 调整到哪一个组.
    gt=Find group where vb belong;
    gt=gt−{vb};
    //在所有的组中,查找满足组内所有节点与 vb 没有隐藏关系的分组.
    found=false;   //先假定没有找到合适的分组
    For each g in G
        if g=gt continue;
        if vb 与 g 组内的成员没有隐藏关系 then
            g=g∪{vb};   //将 vb 放在 g 组中
            Schedule(0,send(vb,group_adjust,g));   //立即通知 vb 节点加入的组为 g
            found=true;
            break;
        if not found then
            G=G∪{{vb}};   //新建一个组,并把 vb 放在新建的组
    Merge(G,H);   //合并节点
}
Merge(G,H)   //如果组与组间,内部所有节点没有隐藏关系,则将其进行合并
{   Can_merge=true;
    While (Can_merge)
    {
        Can_merge=false;
```

© 中国科学院软件研究所 http://www.jos.org.cn

```
        For each g_a in G
            For each g_b in G
                If g_a & g_b 内节点都不存在隐藏关系 then
                    g_a=g_a∪g_b;
                    G=G−g_b;
                    Can_merge=true;
                    Schedule;  //通知 g_b 组内的成员,调整分组
        }
    }
```

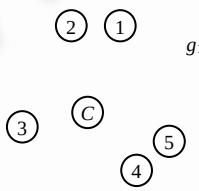
Check_move(*G*,*H*,*v*) //检查与节点 *v* 构成隐藏关系的组,如果组大于 5 个,说明节点 *v* 发生了移动,导致某些隐藏关系不正确,需要清除这些过时信息

```
{   T=与 v 构成隐藏节点的组;
    If |T|>5 then  //如果与 v 构成隐藏节点的组超过 5 个,则认为 v 是移动的节点
    H=H−{v,*};  //清除与 v 构成隐藏关系的历史记录
}
```

下面以实例说明我们的解析策略.

1.2.1 节点静止的情况下

如图 4 所示假设,该网络有 5 个节点并存在 3 个隐藏节点组,分别是{{1,2},{3},{4,5}}.在初始阶段,动态分组算法将这 5 个节点划分到一个竞争组 G={{1,2,3,4,5}},H=∅.



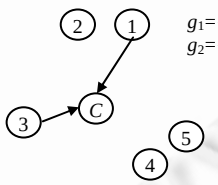
g_1={1,2,3,4,5}

节点	组号	隐藏节点
1	1	
2	1	
3	1	
4	1	
5	1	

Fig.4 Hidden relationships of nodes

图 4 节点隐藏关系

假设协调器节点发现了节点 1 和节点 3 为一对隐藏节点,则根据动态调整算法,对节点 3 进行调整,将其从 g_1 中拆分出来,G={{1,2,4,5},{3}},H={{1,3}}.如图 5 所示.



g_1={1,2,4,5}
g_2={3}

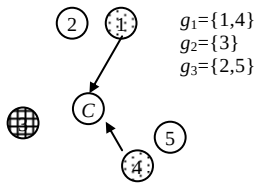
节点	组号	隐藏节点
1	1	3
2	1	
3	2	1
4	1	
5	1	

Fig.5 Dynamically adjusting of nodes

图 5 节点动态调整

经过一段时间的运行后,假设分组情况为 G={{1,4},{3},{2,5}},H={{3,1},{3,2},{3,4},{3,5},{2,4},{5,1}},如图 6 所示.

© 中国科学院软件研究所 http://www.jos.org.cn

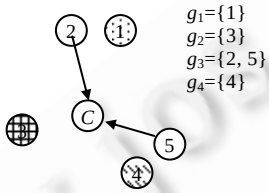


节点	组号	隐藏节点
1	1	3 5
2	3	3 4
3	2	1 2 4 5
4	1	2 3
5	3	1 3

Fig.6 Results after adjustment node running

图 6 运行后节点调整情况

这时,如果节点 1 和节点 4 产生隐藏冲突,根据算法需要将节点 4 从 g_1 中拆分出来,算法先检查 g_2={3},$g2$ 内所有的节点与节点 4 存在隐藏关系,不能加入;再检查 g_3={2,5},尽管节点 4 与节点 5 不存在隐藏关系,但节点 4 与节点 2 存在隐藏关系,节点 4 也不能加入到 g_3 组,因而新建一个组 g_4,并把节点 4 放入组 g_4,如图 7 所示.

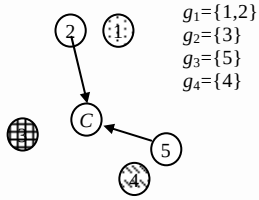


节点	组号	隐藏节点
1	1	3 4 5
2	3	3 4
3	2	1 2 4 5
4	4	1 2 3
5	3	1 3

Fig.7 Results after node adjusting again

图 7 节点再次调整情况

如果节点 5 和节点 2 再产生隐藏冲突,则将节点 2 拆分出来,发现节点 2 可以加入组 g_1.经过以上过程,一共被划分为 4 个组,组内节点不存在隐藏关系.但在算法的最后一个步骤,会对组进行合并检查,发现 g_3 和 g_4 的节点并不存在隐藏关系,可以将其合并,最终结果如图 8 所示.



节点	组号	隐藏节点
1	1	3 4 5
2	1	3 4 5
3	2	1 2 4 5
4	4	1 2 3
5	3	1 2 3

Fig.8 Results of node merging

图 8 节点合并情况

经过以上过程,一共被划分为 3 组{{1,2},{3},{4,5}},组内节点不存在隐藏关系.如图 9 所示.

节点	组号	隐藏节点
1	1	3 4 5
2	1	3 4 5
3	2	1 2 4 5
4	3	1 2 3
5	3	1 2 3

Fig.9 Results of node grouping

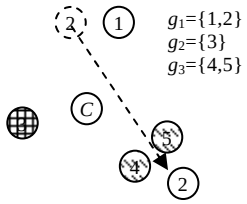
图 9 节点分组情况

下面我们将讨论节点移动的情况.

© 中国科学院软件研究所 http://www.jos.org.cn

1.2.2 节点移动的情况下

假设上图 9 中的节点 2 从顶部移动到底部,如图 10 所示.



节点	组号	隐藏节点
1	1	3 4 5
2	1	3 4 5
3	2	1 2 4 5
4	3	1 2 3
5	3	1 2 3

Fig.10 Results of node moving

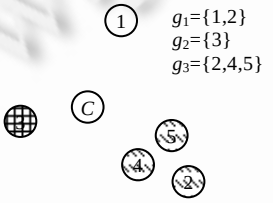
图 10 节点移动

在移动完成后,节点 2 与节点 1 仍在同一个竞争组 1 中,但事实上,节点 1 与节点 2 构成隐藏关系,如果在同一竞争周期发送数据会导致冲突,被动态分组算法识别出来,这时出现两种情况:

(1) 与节点 2 构成隐藏关系的组不超过 5 个,则根据动态分组算法将节点 2 划分在一个新的组;

(2) 超过 5 个,根据文献[16]的证明,一个节点的周围最多只有 5 个隐藏关系组,可以判断节点 2 是移动节点,则将隐藏关系表中与节点 2 相关的隐藏关系清除掉,重新按照动态算法进行调整.

图 11 为节点移动情况下的最终结果.



节点	组号	隐藏节点
1	1	2 3 4 5
2	1	1 3
3	2	1 2 4 5
4	3	1 3
5	3	1 3

Fig.11 Final steady state

图 11 最终稳定状态

与其他分组方法不同,我们采用的是一种动态的、自适应的分组策略.节点分组没有额外的隐藏关系搜索过程,不需要借助额外的广播帧来获取隐藏关系.我们的分组策略是在节点间数据通信过程中动态进行的,根据通信的状况(如节点的加入、移除、位置改变等情况)进行自适应的调整,因而 HNCADS 更贴近实际应用.

2 算法收敛性分析

定理 1. 任意给定一个网络图 T、隐藏关系序列 H 和分组 G,Merge 算法是收敛的.

证明:Merge 算法根据 H 对分组 G 进行两两合并,存在以下两种情况:

(1) 如果对于任意的 $g_i \in G$,无法找到一个组 $g_j \in G$ 且 $g_j \neq g_i$,使得对于任意的$\{v_i,v_j\} \notin H$(其中,$v_i \in g_i, v_j \in g_j$),算法结束;

(2) 反之,G 中存在这样的组 g_j,则将 g_i 与 g_j 合并为一个新的组 g_i',并将原来的 g_i 和 g_j 删除,所得到的新组为 G',$|G'|=|G|-1$.

对于情况(1),Merge 算法继续对 G'进行上述操作,直到情况(2).

综上所述,Merge 算法是收敛的. □

定理 2. 任意给定一个图 $T=\{v_1,v_2,\ldots,v_n\}$、隐藏关系序列 H 和分组 G,动态分组算法 HNCADS 的结果是收敛的.

证明:假定对于图 T,其隐藏关系是逐渐暴露的,即 H 逐渐增长,对于一个给定的图,其隐藏关系的结果是唯一

© 中国科学院软件研究所 http://www.jos.org.cn

的,因此 H 的值是确定的.用归纳法进行证明,假设在迭代过程中,隐藏关系序列为 H_i,相应的分组结果为 $G_i(i=1,2,\ldots,n)$:

(1) 在初始阶段 $H_0=\Phi$,即节点间没有隐藏关系,所有节点在同一个分组即 $G=\{\{v_1,v_2,\ldots,v_n\}\}$;

(2) 假设 H_n 时,动态分组算法收敛的,即在输入的隐藏序列集合为 $H_n=\{h_1,h_2,\ldots,h_k\}$时,动态分组算法 HNCADS 得到的分组结果为 $G_n=\{g_1,g_2,\ldots,g_k\}$,对于任意的 $v_i,v_j\in g_t(1\leqslant t\leqslant k),\{v_i,v_j\}\notin H_n$;

(3) 在 $H_{n+1}=H_n\cup\{\{v_a,v_b\}\}$的情况下,即 v_a 和 v_b 发生隐藏冲突.由于不同分组在不同的时间片发送数据,因此 v_a 与 v_b 必然属于同一个组,设 $v_a,v_b\in g_t$,根据动态分组算法,需要将 v_b 放到别的组.如果 v_b 与 G 中的某一个组内所有成员没有冲突,则将 v_b 节点放在该分组中,调整后的结果为:

$$G'=\{g_1,g_2,\ldots,g_i\cup\{v_b\},\ldots,g_t-\{v_b\},\ldots,g_k\}.$$

如果 v_b 无法加入 G 中现有的组,将 v_b 放在新的组,调整后的结果为

$$G'=\{g_1,g_2,\ldots,g_t-\{v_b\},\ldots,g_k,\{v_b\}\}.$$

最后,算法对调整后的结果 G'进行合并,根据定理 1,其合并得到的结果是收敛的.

综上所述,动态分组算法 HNCADS 的结果是收敛的. □

由定理 1 和定理 2 可知,对于一个给定的网络,如果节点是固定的,节点间的隐藏关系恒定,则运行动态分组算法可以给出一个唯一的分组结果.

定理 3. 任意给定一个图 $T=\{v_1,v_2,\ldots,v_n\}$、隐藏关系序列 H,动态分组算法 HNCADS 的最少调用次数为 1,最大调用次数为$|H|$.

证明:设图有 n 个节点,其隐藏关系为$\{\{v_2,v_1\},\{v_3,v_1\},\ldots,\{v_n,v_1\}\}$,即 v_1 与其他节点构成隐藏关系,而 $v_2,\ldots,v_n$ 间不存在隐藏关系.由于 HNCADS 算法与隐藏关系对内部节点的先后顺序有关,如果输入为$\{v_2,v_1\}$,根据该算法,将 v_1 拆分出来,$\{v_2,v_3,\ldots,v_n\}$保持在同一组,这时,$(v_2,v_3,\ldots,v_n)$已不存在隐藏关系,算法结束,调用次数为 1.而如果输入为$\{v_1,v_2\}$,则算法将 v_2 拆分出来,$\{v_1,v_3,v_4,\ldots,v_n\}$保持在同一个组.这时,$v_1$ 仍然会与$\{v_3,v_4,\ldots,v_n\}$冲突.以上过程如此反复,直到所有隐藏序列全部暴露出来.因此,动态分组算法调用的最大次数为$|H|$.

由定理 3 可知,动态分组算法最多只需运行$|H|$次就结束. □

3 实验仿真

3.1 仿真实验环境

我们采用 NS2 对 802.15.4 标准协议与 HNCADS 改进协议进行仿真与评估比较,其中,802.15.4 协议仿真模型为 WPAN[20],该模型支持信标使能.我们在其基础上添加 HNCADS 策略,超码框的 CAP 周期根据分组数均匀划分,每组独享一个时间片.

具体的实验环境如下:采用信标使能的 802.15.4 星型网络,并假设无线信道无噪声且不引入错误模型,网络中一共有 19 个节点,分布在半径为 10 米的圆形区域内,其中,id=0 的为 PAN 协调器,其余 18 个为固定的普通节点(id=1,2,...,18),每个节点的信号传输半径和感知半径为 15 米,即所有节点可以与协调器节点进行相互通信,但部分普通节点间存在隐藏关系(节点间的间距大于节点的感知半径),例如图 12 的节点 2 与节点 5 以及节点 12 与节点 8 等.

在仿真分析中需要用到协议中定义的参数常量有:

- *aNumSuperframeSlots*=16;
- *aBaseSlotDuration*=0.96ms;
- *aBaseSuperframeDuration*=15.36ms.

这 3 个参数定义了自由竞争周期的长度和基本时槽的长度.同时,将设置超帧参数 $BO=SO=3$,BO 和 SO 在信标使能网络中都相同的值,就意味着在超帧中也没有采用休眠周期,即帧 $BI=SD$=122.88ms.仿真实验中所用到的大部分参数都是根据 IEEE 802.15.4 标准来设定的.

© 中国科学院软件研究所 http://www.jos.org.cn

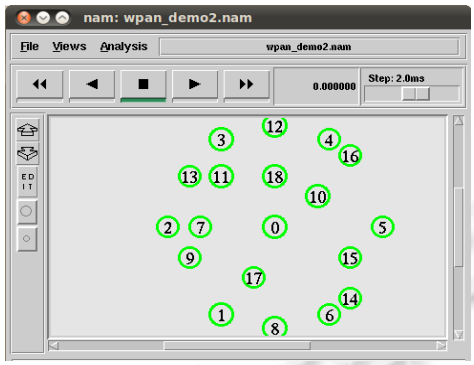


Fig.12 The scene of simulation nodes

图 12 仿真节点分布

3.2 性能评价指标

在 IEEE 802.15.4 MAC 层中,影响网络性能的因素有很多,包括网络参数(网络规模、拓扑等)、节点参数(缓冲区大小、传输和侦听距离等),还有流量参数等.为了全面地评估 IEEE 802.15.4 和改进协议 HNCADS 在各种仿真环境下的性能,本文选择以下的 4 个性能指标:包投递率、端到端时延、网络吞吐量、能量消耗.

1. 包投递率

包投递率是指成功收到的数据帧个数与发送数据帧个数之比,为了去除上层对 IEEE 802.15.4 的影响,这里不区别直接发送成功和重发成功的情况.只有被永久丢弃的数据包,才被计入丢包数量中,并只记录为 1 次,不累计此包之前被重复丢弃的次数.若数据包最终被成功接收,即使之前被反复的丢弃重传,也不计入丢包数量.如此可以保证发送的数据包数量等于接收和丢弃数据包数量.

2. 端到端时延

端到端时延指在应用层数据包从发送方发出到接收方收到的时间差.本文关注的是重传延时,主要是指由于超时或冲突造成的帧重传而延误的时间.文中采用统计平均的方法对不同负载下的网络时延进行评估:

$$Delay = \sum_{0,1...,total} Delay_{frame} / total,$$

其中,*total* 为成功接收数据帧的个数.

3. 网络吞吐量

网络的吞吐量是网络性能的一个重要参数,是指在不丢帧的情况下,单位时间内网络能够传输数据帧的最大数量.本文对整个仿真过程中网络吞吐量的实时变化进行统计,采用的计算公式为

$$Throughput=(sum[i]+last\_throughput)/granularity\times 10^6,$$

其中,*sum*[*i*]为当前统计间隔内成功传输的无差错字节数;*last_throughput* 为前一个统计间隔内成功传输字节数的平均值;*granularity* 为统计间隔,本文取 100 个超帧周期为一个统计间隔,采用的吞吐量单位为 Mb/s.

4. 能量消耗

由于 IEEE 802.15.4 最突出的特性就是低功耗,所以能量消耗也是评价网络性能的一个重要指标.根据本课题采用的仿真模型,以下公式可以实现对网络中所有具有数据传输的节点进行能耗计算:

$$Power\_consume_{node}=P_{tx}\times T_{tx}+P_{rx}\times T_{rx}+P_{idle}\times T_{idle},$$

其中,P_{tx},P_{rx} 和 P_{idle} 分别代表节点在发送、接收和空闲状态所消耗的能量,单位为焦耳;T_{tx},T_{rx} 和 T_{idle} 分别代表节点在发送、接收和空闲状态所花费的时间.本文将用节点初始能量减去消耗能量,求得节点在仿真结束后的剩余能量,以此来衡量各种网络环境下的能耗情况.

3.3 实验结果

为了评估协议的性能,我们让所有节点在同一时刻 10 开始往节点 0 发送数据包,在时刻 100 时,随机选择 4

© 中国科学院软件研究所 http://www.jos.org.cn

个节点让其在圆形区域内随机移动,直到时刻 200 结束,分别计算 IEEE 802.15.4 时隙协议、FDGS 协议[16]和我们的改进协议(HNCADS)的包投递率、端到端时延、网络吞吐量、能量消耗.节点发送数据的时间间隔从 0.2pps~1pps,增幅为 0.2,用于模拟不同负载情况下协议的性能,每组数据实验 20 次,结果取平均值.具体的实验结果如图 13 所示.

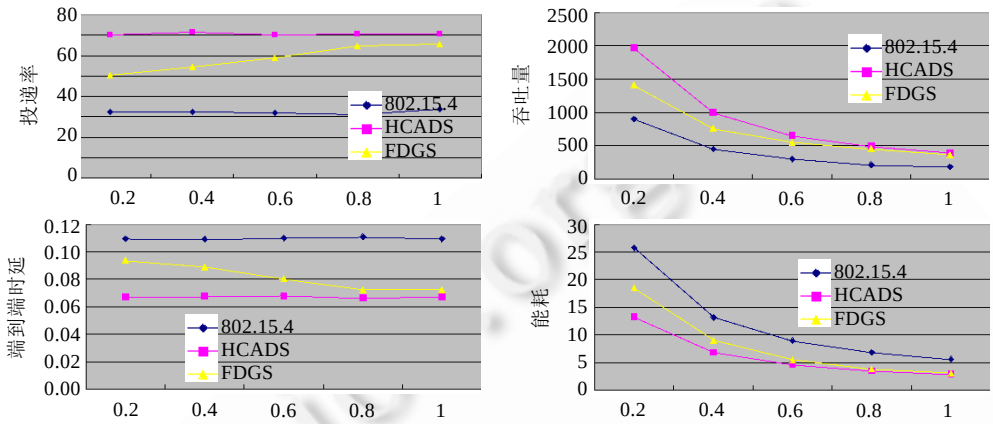


Fig.13 The results of experiment

图 13 实验结果

从图 13 左上图来看:改进前,IEEE 802.15.4 的投递率较低,大约为 32.1%;而 HNCADS 数据包投递率则稳定在 70.5%,为改进前的 2.19 倍;而 FDGS 协议的投递率介于二者之间,在低发包频率情况下,投递率达到 65.7%,但在高发包频率情况下则下降到 50.3%.这是由于 IEEE 802.15.4 MAC 没有任何可以防范隐藏节点问题的机制,冲突发生后,数据包在重传多次失败后被永久丢弃,因而导致数据包投递率较低;而改进的 HNCADS 协议则大幅减少了隐藏冲突,因而数据包投递率大幅提高;而 FDGS 协议尽管做了分组,但由于在节点移动的情况下分组信息不能静态调整,因而到达率低于能够动态进行分组调整的 HNCADS.

从图 13 右上图来看:IEEE 802.15.4,FDGS,HNCADS 协议的网络吞吐量均随着发包频率的上升而提高.在低发包频率的情况下(1PPS),网络吞吐量分别 0.18Mb/s,0.37Mb/s,0.39Mb/s.后两种协议的吞吐量约为 IEEE 802.15.4 时隙协议的 2 倍,在高发包频率的情况下(5PPS),网络吞吐量分别为 0.9Mb/s,1.41Mb/s,1.97Mb/s.这是因为后两种协议均采用了分组策略,能够有效减少网络的隐藏冲突,提高网络吞吐量.但受部分节点移动隐藏关系发生改变的影响,FDGS 因不能及时调整分组,导致网络依然存在隐藏冲突,网络吞吐量的表现略低于 HNCADS 协议.

从图 13 左下图来看:IEEE 802.15.4 和 HNCADS 的端到端延时分别为 0.11s 和 0.07s;而 FDGS 在低发包频率情况下,端到端延时与 HNCADS 大致一样,但在高发包频率下,端到端延时则上升到 0.09s.这是由于 FDGS 和 HNCADS 采用了分组机制,极大地减少了隐藏冲突,因而节点因隐藏冲突而进行二进制指数后退回避的时间极大地减少,端到端延时也就相应得到显著提高.

从图 13 右下图来看:IEEE 802.15.4,FDGS,HNCADS 协议的的平均能耗随着发送频率的增加而增加,在低发包频率的情况下,三者平均能耗大致相当;但在高发包频率的情况下,FDGS 和 HNCADS 均由于 IEEE 802.15.4 协议.这是因为在低发包频率情况下,网络发生隐藏冲突和竞争冲突的概率很低,数据发送成功率高,因而能耗较少.而在高发包频率情况下,网络发生隐藏冲突和竞争冲突的概率增加,冲突重传的次数增加,能耗随着增大.

从图 13 可以看出,在 IEEE 802.15.4 信标使能的星型网络中,当网络中节点存在隐藏关系以及部分节点移动的情况下时,IEEE 802.15.4 中 CSMA/CA 信道接入机制不能体现良好的性能,频繁的冲突和重发不可避免,网络性能大大降低,能耗也不断增加.当采用了分组机制后,网络在包投递率、端到端时延、吞吐量、能耗等性能指

© 中国科学院软件研究所 http://www.jos.org.cn

标上得到明显的改善.而 HNCADS 采用动态调整方式,分组能够随着节点的移动而动态调整,网络的性能最优.

4 总 结

隐藏节点问题是 IEEE 802.15.4 网络普遍存在的问题,在网络节点数量较多或节点数据发送频率较高的情况下,会严重影响网络的吞吐率和消耗节点能量.本文提出的 HNCADS 可以彻底消除隐藏节点问题,大幅提高网络性能.与其他隐藏冲突解析机制相比,具有能够在节点移动的场景下自适应调整分组的优点.

References:

[1] Xie R, Qi DY, Li YJ. A novel distributed MCDS approximation algorithm for wireless sensor networks. Wireless Communications & Mobile Computing, 2009,9(3):427−437. [doi: 10.1002/wcm.547]

[2] Gu JJ, Chen SC, Zhuang Y. Wireless sensor network-based topology structures for the Internet of things localization. Chinese Journal of Computers, 2009,33(9):1458−1556 (in Chinese with English abstract). [doi: 10.3724%2fSP.J.1016.2010.01548]

[3] Tseng YC, Ni SY, Shih EY. Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network. IEEE Trans. on Computers, 2003,52(5):545−557. [doi: 10.1109/TC.2003.1197122]

[4] Li YJ, Wang ZJ. Cost analysis and optimization for IP multicast group management. Computer Communications, 2007,30(8): 1721−1730. [doi: 10.1016/j.comcom.2007.02.002]

[5] Tobagi A, Kleinrock L. The hidden terminal problem in carrier sense multiple-access and the busy-tone solution. IEEE Trans. on Communication, 1975,23(12):1417−1433.

[6] Koubaa A, Severino R, Alves M, Tovar E. Improving quality-of-service in wireless sensor networks by mitigating hidden-node collisions. IEEE Trans. on Industrial Informatics, 2009,5(3):299−313. [doi: 10.1109/TII.2009.2026643]

[7] Haas ZJ, Deng J. Dual busy tone multiple access (DBTMA)-a multiple access control scheme for ad hoc networks. IEEE Trans. on Communications, 2002,50(6):975−985.

[8] Alshanyour A, Agarwal A. Performance of IEEE 802.11 RTS/CTS with finite buffer and load in imperfect channels: Modeling and analysis. In: Proc. of the 2010 IEEE Global Telecommunications Conf. Florida, 2010. 6−10. [doi: 10.1109/GLOCOM.2010.5684057]

[9] Leng S, Zhang L, Chen Y. IEEE 802.11 MAC protocol enhanced by busy tones. In: Proc. of the 2005 IEEE Int'l Conf. on Communications. Seoul, 2005. 2969−2973. [doi: 10.1109/ICC.2005.1494936]

[10] Zhai H, Fang Y. A solution to hidden terminal problem over a single channel in wireless ad hoc networks. In: Proc. of the MILCOM 2006. Washington, 2006. 1−7. [doi: 10.1109/MILCOM.2006.302394]

[11] Eisenman SB, Campbell AT. Structuring contention-based channel access in wireless sensor networks. In: Proc. of the 5th Int'l Conf. on Information Processing in Sensor Networks. Nashville, 2006. 226−234. [doi: 10.1109/IPSN.2006.243751]

[12] Jung ES, Vaidya NH. A power control MAC protocol for ad hoc networks. In: Proc. of the ACM MobiCom 2002. Westin Peachtree Plaza, 2002. 36−47. [doi: 10.1145/570645.570651]

[13] Li YJ, Chen H, Xie R. A novel scatternet formation algorithm for bluetooth-based sensor networks. Mobile Information Systems, 2011,7(2):93−106.

[14] Sheth A, Han R. SHUSH: Reactive transmit power control for wireless MAC protocols. In: Proc. of the 1st Int'l Conf. on Wireless Internet. Budapest, 2005. 18−25. [doi: 10.1109/WICON.2005.29]

[15] Zhao BH, Zhang W, Liu HC, Qu YG. Cluster partition algorithm in wireless sensor networks. Chinese Journal of Computers, 2006,29(1):161−165 (in Chinese with English abstract).

[16] Hwang LJ, Sheu ST, Shih YY, Cheng YC. Grouping strategy for solving hidden node problem in IEEE 802.15.4 LR-WPAN. In: Proc. of the 1st Int'l Conf. on Wireless Internet. Budapest, 2005. 26−32.

[17] Zhang XB, Cheng LL, Zhu QM. SAHRC: A cluster-based routing control protocol for wireless sensor network. Journal of Electritic & Information, 2011,33(8):2013−2017 (in Chinese with English abstract). [doi: 10.3724/SP.J.1146.2010.01270]

[18] Kwon CH, Tek RJ, Kim KH, Yoo SH. Dynamic group allocation scheme for avoiding hidden node problem in IEEE 802.15.4. In: Proc. of the 2009 Int'l Symp. on Communications and Information Technologies. Incheon, 2009. 637−638. [doi: 10.1109/ISCIT.2009.5341168]

© 中国科学院软件研究所 http://www.jos.org.cn

[19] Sheu ST, Shih YY. P-Frozen contention strategy (PFCS) for solving collision chain problem in IEEE 802.15.4 WPANs. In: Proc. of the 2006 IEEE 63rd Vehicular Technology Conf. Melbourne, 2006. 1323−1327. [doi: 10.1109/VETECS.2006.1683049]

[20] Zheng JL, Lee MJ. A resource-efficient and scalable wireless mesh routing protocol. Ad Hoc Networks, 2007,5(6):704−718. [doi: 10.1016/j.adhoc.2006.11.003]

附中文参考文献:

[2] 顾晶晶,陈松灿,庄毅.基于无线传感器网络拓扑结构的物联网定位模型.计算机学报,2009,33(9):1458−1556. [doi: 10.3724/SP.J.1016.2010.01548]

[15] 赵保华,张炜,刘恒昌,屈玉贵.无线传感器网络中的组划分算法.计算机学报,2006,29(1):161−165.

[17] 张小波,程良伦,Zhu Quan-Min. SAHRC:一种基于分簇的无线传感器网络路由控制算法.电子与信息学报,2011,33(8):2013−2017. [doi: 10.3724/SP.J.1146.2010.01270]

李拥军(1968－),男,湖南邵东人,博士,教授,主要研究领域为计算机网络,分布计算.
E-mail: liyj@scut.edu.cn

谢嵘(1974－),男,博士,副教授,主要研究领域为计算机网络性能分析与仿真,新一代网络技术,IP 多播,无线传感器网络.
E-mail: rong_xie@126.com

谭晓青(1976－),女,博士,副教授,主要研究领域为密码编码学,信息安全.
E-mail: ttanxq@jnu.edu.cn

© 中国科学院软件研究所 http://www.jos.org.cn