

# 基于地址空间的树型网络地址分配

刘宁宁<sup>1</sup> 樊建席<sup>1</sup> 林政宽<sup>2</sup>

1 苏州大学计算机科学与技术学院 江苏 苏州 215006

2 福州大学数学与计算机科学学院 福州 350108

(20175427005@stu.suda.edu.cn)



**摘要** 无线传感器网络(WSN)是由大量部署在监测区域内的微型传感器节点通过无线通信方式组成的一个多跳、自组织的网络。分布式的环境感知能力和简单灵活的部署方式,使得WSN成为影响人们日常生活的重要因素;并且随着微电子技术和通信技术的不断发展,WSN已被广泛应用于国防军事、环境监测、医疗健康、智能家居和工业制造等领域。ZigBee是一种支持低速率传输、低功耗、安全可靠的面向可用产品及应用的无线个人局域网的全球标准,与蓝牙、Wi-Fi等其他无线个人局域网标准不同,其提供了低功耗的无线树状和网状网络,可以支持上千个无线传感器设备在网络中使用。ZigBee技术的分布式地址分配机制(Distributed Address Assignment Mechanism, DAAM)中存在网络孤立节点,这种情况导致闲置地址无法使用,且造成了资源浪费。针对这一问题,提出了一种新的树型网络地址分配和路由算法(Address Assignment Algorithm for Tree Network, AAN),在保持与原有协议兼容的基础上通过协调器节点对网络进行维护和控制,各节点根据算法设定的步骤依次进行地址空间分配。该算法可以减少网络中的闲置地址空间及网络中的孤立节点数,优化网络拓扑结构,减少建立与维护路由表所需的时间与存储空间。仿真实验结果表明,所提算法在地址分配成功率、孤立节点数以及网络深度方面优于DAAM算法。

**关键词:** 无线传感器网络; ZigBee; 树型网络; 地址分配; DAAM

**中图法分类号** TPO243

## Address Assignment Algorithm for Tree Network Based on Address Space

LIU Ning-ning<sup>1</sup>, FAN Jian-xi<sup>1</sup> and LIN Cheng-kuan<sup>2</sup>

1 School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215006, China

2 College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China

**Abstract** Wireless sensor network (WSN) is a multi-hop and self-organizing network composed of a large number of micro-sensor nodes deployed in the monitoring area through wireless communication. Distributed environment awareness and simple and flexible deployment make WSN become an important factor affecting our daily life. With the continuous development of micro-electronics and communication technology, WSN has been widely used in national defense, military, environmental monitoring, medical health, smart home and industrial manufacturing. ZigBee is a global standard for wireless personal area networks that support low-rate transmission, low power consumption, security and reliability for available products and applications. Different from the other wireless personal area network standard such as Bluetooth, Wi-Fi, ZigBee provides the low power wireless tree and mesh networking, and supports up to thousands of wireless sensor devices in a network. There exist isolated nodes in the Distributed Address Assignment Mechanism of ZigBee technology, which results in unavailable of idle address and waste of resources. To solve this problem, a novel tree-based Address Assignment Algorithm for Tree Network (AAN) is proposed in this paper. This algorithm can decrease the idle address space and the number of isolated nodes in the network, optimize the network topology, and reduce the time and storage space needed to establish and maintain the routing table. Simulation experiment results show that our algorithm is more advantage than DAAM and one of its present improvements in terms of the success rate of address assignment, number of isolated nodes, and network depth.

**Keywords** Wireless sensor network, ZigBee, Tree network, Address assignment, DAAM

收稿日期:2019-04-24 返修日期:2019-09-16 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金(61572337,61872257)

This work was supported by the National Natural Science Foundation of China (61572337,61872257).

通信作者:林政宽(cklin@suda.edu.cn)

## 1 引言

近年来,随着网络和通信技术的快速发展,采用 ZigBee 标准<sup>[1]</sup>的无线传感器网络<sup>[2]</sup>得到了社会各界越来越多的关注。ZigBee 协议有两种最主要的地址分配方式,分别是分布式地址分配机制和随机地址分配机制(Stochastic Address Assignment Mechanism, SAAM)。DAAM 地址分配方式将“地址-位置”的对应关系包含其中,为树的路由协议<sup>[3]</sup>的运行提供了便利,路由的相关信息可以通过对地址的计算获得。同时,分布式地址分配方式呈现规律性。由于实现方式较为简单,分布式地址分配方式已经得到了较为广泛的研究和应用。DAAM 在组网前必须先决定  $G_{\max}$ ,  $R_{\max}$  与  $L_{\max}$  这 3 个参数,由于这些参数没有针对实际网络结构进行优化,导致实际应用时容易发生闲置地址无法使用的问题,形成了大量的孤立节点<sup>[4]</sup>。SAAM 相对较为简单,只能由已经在网络中的节点为准备加入该网络中的节点随机地分配网络地址,但是这样就导致了地址冲突的问题,并且检测耗时长、投入大。

本文提出了一种新的树型网络地址分配算法,该算法除了能够解决 DAAM 中存在的问题外,还能够优化网络拓扑结构,减少网络中的孤立节点数,灵活地分配网络地址空间,提高节点间地址分配的效率和新生节点的入网成功率。与此同时,本文也提出了一种基于上述算法的路由方式,相较于传统的路由方法,新的路由方法不需要通过路由表进行信息传输,类似于 DAAM 的方式,封包在节点的转发可以直接根据目的地地址经过计算得到。因此,这种方式可以加快路由,减少路由表的建立和维护所花费的时间和存储空间。

## 2 相关工作

迄今为止,很多研究人员已经对网络的地址分配算法进行了大量的科学研究,并且取得了诸多科研成果<sup>[5-6]</sup>。但是,物联网近年的蓬勃发展造成了网络规模大,使得网络拓扑结构具有动态性,因此关于无线传感器网络地址分配算法的研究仍有不足之处,需要从多个方面对地址分配算法进行改进和提高<sup>[7-8]</sup>。特别是 ZigBee 协议中的 DAAM 地址分配方式,其虽然能够快速分配地址,且具有良好的路由协议,但是存在地址空间分配不均和容易产生孤立节点的问题。所谓的孤立节点问题,指的是网络中有着空闲的地址空间,但因地址分配方法的限制造成部分节点无法连接到网络,这些节点就称为孤立节点。孤立节点的问题在 DAAM 算法中尤其严重。

针对子空间不足导致路由节点无法获得地址空间的问题,Giri 等<sup>[9]</sup>与 Fang 等<sup>[10]</sup>分别提出了在不改变组网参数的状态下向其他节点借用地址的分配算法,即分配地址空间不足的路由节点可以向有剩余地址的路由节点借地址后再分配,这有助于实现网络负载的均衡,但是存在着额外通信开销和破坏原有树型路由协议的问题。针对 DAAM 中产生大量孤立节点的问题,Li 等<sup>[11]</sup>提出了一种新的方法,该算法主要使用 DAAM 未用到的地址,对其进行重新分配,通过地址重配置减少孤立节点。但是,理论分析表明这种可扩展操作只

是一次性的,同时重分配速度慢且效率低下。Gami 等<sup>[12]</sup>针对可分配地址宽度不足的问题,提出了 SLAR(Single Level Address Reorganization)方案,通过增大深度参数来减小地址偏移量,从而增加了可分配的地址空间。这种以深度换取宽度的方法虽然可取,但是一旦减小网络的深度,地址重配置的操作就会使得控制开销和耗时明显增加。Lorincz 等<sup>[13]</sup>使用集中式的迭代建树,通过分布式的孤立节点主动探测路径和分布式的先深度后宽度搜索方法,减少了孤立节点,但增加了算法的时间复杂度和组网开销。Ren 等<sup>[14]</sup>提出了基于分段的按需可扩展地址分配算法,并通过按需扩展路由节点的地址空间提高了节点地址空间的使用效率,但这种做法并没有从根本上改变父节点可分配地址不足的问题。近期,针对树型网络拓扑结构中低功耗无线个人局域网使用 IPv6 路由协议标准的数据包向上发送问题,Peres 等<sup>[15]</sup>基于分层 IPv6 地址分配的思想,探索无周期网络拓扑结构,提出了 Matrix 的解决方案。这是一种用于动态网络拓扑且能够容错任何在 6LoWPAN 中的数据流的路由方案,为 6LoWPAN 中的任意路由提供了可靠且可扩展的解决方案;但这种方法并未考虑到节点的移动性,移动节点数量越多内存占有量越大。

## 3 AAN 算法

目前,树型网络的地址分配方法主要以 ZigBee 网络中基于 DAAM 的路由算法为基础,它使用网络协调器和路由节点的最大子节点数  $C_{\max}$ 、最大子路由由节点数  $R_{\max}$  和网络最大深度  $L_{\max}$  这 3 个参数来计算并分配地址空间。该算法定义了  $C_{skip}(d)$  函数来表示地址偏移量,为每个节点分配网络地址,网络中的协调器及所有的路由节点将自己所具有的地址块平均分为  $R_{\max} + 1$  块,将前面的  $R_{\max}$  个地址块平均分配给连接到自己的路由节点<sup>[16]</sup>,将后面的最后一个地址块分配给  $C_{\max} - R_{\max}$  个连接到自身的终端节点<sup>[17]</sup>。地址空间大小的计算式如下:

$$C_{skip}(d) = \begin{cases} 1 + C_{\max} \times (L_{\max} - d - 1), & R_{\max} = 1 \\ \frac{1 - C_{\max} \times R_{\max}^{(L_{\max} - d - 1)} - C_{\max} - R_{\max}}{1 - R_{\max}}, & R_{\max} \neq 1 \end{cases}$$

其中, $d$  表示当前节点的深度。

如果一个网络节点的  $C_{skip}(d)$  为 0,说明它将不会有地址偏移量,即没有可用于分配的地址块,因此其只能作为逻辑上的叶子节点。同样地,如果  $C_{skip}(d)$  大于 0,那么它可以允许子节点关联入网,并分配地址。简单举例,如图 1 所示,协调器节点将总的地址空间  $\{0, 1, 2, \dots, 1200\}$  平均分成 3 段并分别分给 3 个路由节点,每一个路由节点可获得 400 个地址,其可分配的地址分别为  $\{1, 2, \dots, 400\}$ ,  $\{401, 402, \dots, 800\}$  和  $\{801, 802, \dots, 1200\}$ 。每个路由节点又将分配给自己的节点地址子域分配给自己的子节点。在地址分配的过程中发现,由于地址被均匀分配,第一段子域只有 6 个节点,但获得了 400 个地址,而第三段子域的 534 个节点也只获得了 400 个地址,因此导致了地址分配不均与浪费的问题,同时也产生了孤立节点。为了更好地解决该问题,本文设计了一种新的

树型网络地址分配算法(AAN),该方法不仅适用于 ZigBee 网络,还适用于其他的自组织网络。ZigBee 网络最多可容纳 65 536 个无线传感器节点,因此只需要将算法中所使用的可分配地址范围由 0 到 65 535 修改为实际可使用的连续可分配地址空间即可。

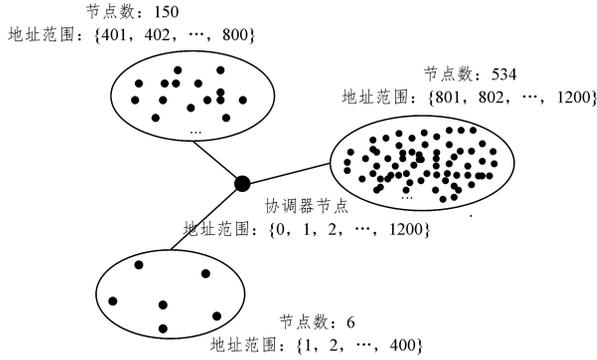


图 1 DAAM 地址分配方式

Fig. 1 DAAM address allocation

本文所建立的树型网络中,节点分为协调器节点、路由节点和终端叶节点 3 种。协调器节点主要负责树型网络建立的初始化工作,以及该网络的维护和控制。同时,协调器节点还决定着  $[a, b], k, R_{\max}$  与  $E_{\max}$  共 4 个参数。其中,  $[a, b] = \{a, a+1, \dots, b\}$  为该树型网络的总可使用地址空间范围,  $k$  为节点统计其  $k$  跳距离内未加入网络的节点的跳数,  $R_{\max}$  表示路由节点的最大个数,  $E_{\max}$  表示终端叶节点的最大个数。令  $u[k]$  表示节点  $u$  的  $k$  跳内未加入网络的节点数。路由节点在网络中负责自身信息的发送与来自其他节点信息的转传。终端叶节点在网络中只可以与协调器节点或路由节点连接,只有与其相连接的协调器节点或路由节点才能进行消息发送,其本身(终端叶节点)不负责来自网络中其他节点的信息转传。网络中的节点称为其分配地址空间的节点为父节点,称接收到其分配地址空间的节点为其子节点。

地址分配实施方法如下:

首先,协调器节点决定该树型网络的总可使用地址范围  $[a, b]$  和  $k, R_{\max}, E_{\max}$  3 个参数,然后网络中的节点按下面的步骤进行地址分配。

步骤 1 当某节点  $z$  所分配到的可使用地址空间为  $[x,$

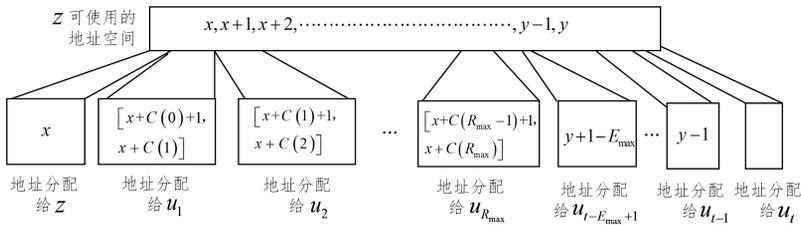


图 3 当  $t > R_{\max} + E_{\max}$  时地址的分配情况

Fig. 3 Address allocation when  $t > R_{\max} + E_{\max}$

情况 1.3  $R_{\max} < t \leq R_{\max} + E_{\max}$

首先,对于所有的  $i \in \{R_{\max} + 1, R_{\max} + 2, \dots, t\}$ , 分配地址空间  $[i + y - t, i + y - t]$  与  $k, R_{\max}, E_{\max}$  给节点  $u_i$ 。

然后,令  $c_0 = 0$  且  $c_1 = y - x - t + R_{\max} - \sum_{j=2}^{R_{\max}} c_j$ , 对于所有的

$y]$  时,若  $y - x \geq 1$ , 则执行步骤 2; 否则执行步骤 4。

步骤 2 网络中的节点可以通过广播的方式询问其一跳内未加入网络的节点是否加入该网络,以计数的方法统计这些节点并回复其跳内未加入网络的节点数。

步骤 3 当节点  $z$  收集完请求加入的节点回复后,令  $\{u_1, u_2, \dots, u_t\}$  为请求加入的节点集合,其满足对于所有的  $i \in \{1, 2, \dots, t-1\}$ , 有  $u_i[k] \geq u_{i+1}[k]$ 。然后,根据下述的子步骤进行地址空间分配。

情况 1  $t \leq y - x$

情况 1.1  $t \leq R_{\max}$

令  $c_0 = 0$  且  $c_1 = y - x - \sum_{j=2}^t c_j$ 。对于所有的  $i \in \{2, 3, \dots, t\}$ , 令  $c_i = \left\lfloor \frac{u_i[k] \times (y - x)}{\sum_{j=1}^i u_j[k]} \right\rfloor$ 。对于所有的  $i \in \{1, 2, \dots, t\}$ , 分配地址空间  $[x + C(i-1) + 1, x + C(i)]$  与  $k, R_{\max}, E_{\max}$  给节点  $u_i$ , 其中  $C(i) = \sum_{j=0}^i c_j$  (见图 2)。

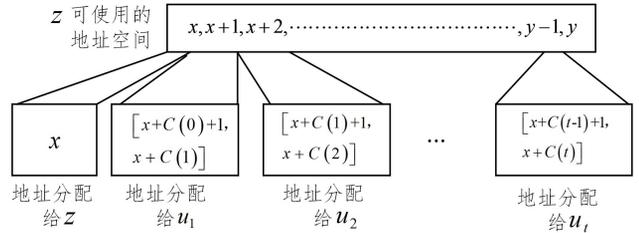


图 2 当  $t \leq R_{\max}$  时地址的分配情况

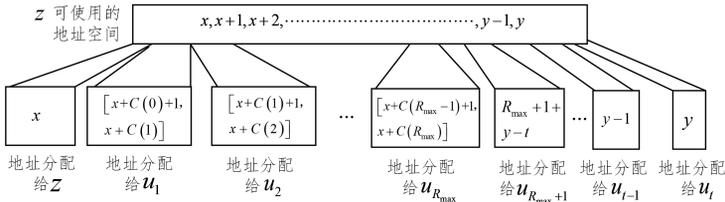
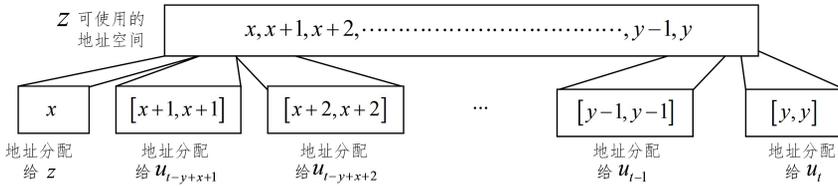
Fig. 2 Address allocation when  $t \leq R_{\max}$

情况 1.2  $t > R_{\max} + E_{\max}$

首先,若  $E_{\max} > 0$ , 则对于所有的  $i \in \{t - E_{\max} + 1, t - E_{\max} + 2, \dots, t\}$ , 分配地址空间  $[i + y - t, i + y - t]$  与  $k, R_{\max}, E_{\max}$  给节点  $u_i$ 。

然后,令  $c_0 = 0$  且  $c_1 = y - x - \sum_{j=2}^{R_{\max}} c_j - E_{\max}$ 。对于所有的  $i \in \{2, 3, \dots, R_{\max}\}$ , 令  $c_i = \left\lfloor \frac{u_i[k] \times (y - x - E_{\max})}{\sum_{j=1}^{R_{\max}} u_j[k]} \right\rfloor$ 。对于所有的  $i \in \{1, 2, \dots, R_{\max}\}$ , 分配地址空间  $[x + C(i-1) + 1, x + C(i)]$  与  $k, R_{\max}, E_{\max}$  给节点  $u_i$ , 其中  $C(i) = \sum_{j=0}^i c_j$  (见图 3)。

对于所有的  $i \in \{2, 3, \dots, R_{\max}\}$ , 令  $c_i = \left\lfloor \frac{u_i[k] \times (y - x - t + R_{\max})}{\sum_{j=1}^{R_{\max}} u_j[k]} \right\rfloor$ 。对于所有的  $i \in \{1, 2, \dots, R_{\max}\}$ , 分配地址空间  $[x + C(i-1) + 1, x + C(i)]$  与  $k, R_{\max}, E_{\max}$  给节点  $u_i$ , 其中  $C(i) = \sum_{j=0}^i c_j$  (见图 4)。

图 4 当  $R_{\max} < t \leq R_{\max} + E_{\max}$  时地址的分配情况Fig. 4 Address allocation when  $R_{\max} < t \leq R_{\max} + E_{\max}$ 情况 2  $t > y - x$ 情况 2.1  $R_{\max} + E_{\max} \geq y - x$ 对于所有的  $i \in \{t - y + x + 1, t - y + x + 2, \dots, t\}$ , 分配地址空间  $[i + y - t, i + y - t]$  与  $k, R_{\max}, E_{\max}$  给节点  $u_i$  (见图 5)。图 5 当  $R_{\max} + E_{\max} \geq y - x$  时地址的分配情况Fig. 5 Address allocation when  $R_{\max} + E_{\max} \geq y - x$ 情况 2.2  $R_{\max} + E_{\max} < y - x$ 首先,若  $E_{\max} > 0$ , 则对于所有的  $i \in \{t - E_{\max} + 1, t - E_{\max} + 2, \dots, t\}$ , 分配地址空间  $[i - t + y, i - t + y]$  与  $k, R_{\max}, E_{\max}$  给节点  $u_i$ 。然后,令  $c_0 = 0$  且令  $c_1 = y - x - \sum_{j=2}^{R_{\max}} c_j - E_{\max}$ 。对于所有的  $i \in \{2, 3, \dots, R_{\max}\}$ , 令  $c_i = \left\lfloor \frac{u_i[k] \times (y - x - E_{\max})}{\sum_{j=1}^{R_{\max}} u_j[k]} \right\rfloor$ 。对于所有的  $i \in \{1, 2, \dots, R_{\max}\}$ , 分配地址空间  $[x + C(i - 1) + 1, x + C(i)]$  与  $k, R_{\max}, E_{\max}$  给节点  $u_i$ , 其中  $C(i) = \sum_{j=0}^i c_j$ 。步骤 4 节点  $z$  结束分配地址空间的任务。

## 4 路由实施方法

某节点收到封包时,依下列情形处理该封包。

情形 1 若该封包的目的地为该节点的地址,则接收该封包。

情形 2 若该封包的目的地在其可使用的地址空间范围内且不是该节点的地址,则将该封包发送至对应地址的子节点。

情形 3 若该封包不在其可使用的地址空间范围内且该节点并不是协调器节点,则将该封包发送至其父节点。若该节点是协调器节点,则舍弃该封包。

情形 4 若该封包不在其可使用的地址空间范围内且该节点是协调器节点,则舍弃该封包。

在上述路由方法中,收到封包的节点只需要自行分析该封包的目的地是否在自己所分配的节点空间内即可。若在,则将该封包转发给相应的子节点或是自行接收该封包;否则就将该封包转发给自己的父节点。相较于传统的封包转发方式而言<sup>[18]</sup>,该方式由于不需要查询路由表,因此可以减少存储路由表所需要使用的存储空间,同时减少了维护路由表所花费的代价。

## 5 仿真实验与性能分析

本文使用 Java 语言在 Eclipse 环境下分别对基于地址空

间的树型网络地址分配、路由算法 AAN 和分布式地址分配机制 DAAM 进行仿真实验,并对实验结果进行了对比分析。首先,从地址分配成功率、孤立节点数、网络最大深度 3 方面进行横向对比;其次,通过改变  $R_{\max}$  参数来进行纵向的实验对比。

### 5.1 仿真参数的设置

在网络覆盖面积为  $300 \text{ m} \times 300 \text{ m}$  时,设置不同节点密度的 6 个仿真实验环境,  $N$  个节点随机均匀地分布在实验环境中,其中  $N \in \{400, 600, 800, 1000, 1200, 1400\}$ ; 每个场景都只有一个协调器节点并且位于网络中心,每个仿真实验环境下分别做 50 次实验,并取 50 次实验结果的平均值。具体实验参数如表 1 所列。

表 1 仿真环境

Table 1 Simulation environment

参数	数值
网络范围/ $\text{m}^2$	$300 \times 300$
节点数量	400, 600, 800, 1000, 1200, 1400
节点通信半径/m	30
$E_{\max}$	8
$R_{\max}$	5
$L_{\max}$	8
$k$	3
部署类型	随机均匀分布
协调器位置	网络中心

### 5.2 仿真结果与分析

#### 5.2.1 地址分配成功率与孤立节点数

地址分配成功率可以用来衡量地址分配算法的有效性。

定义地址分配成功率为  $S = \frac{N_s}{N}$ , 其中,  $N$  表示节点总数,  $N_s$ 表示获得地址的节点数。当  $E_{\max} = 8, R_{\max} = 5, L_{\max} = 8$  且  $k = 3$  时,从图 6 中可以清楚地发现, AAN 在 6 种不同节点网络环境中的地址分配成功率均明显高于 DAAM, AAN 在最好情况下的地址分配成功率可以达到 98.3%, 而 DAAM 在最好情况下仅能达到 77.1%, 两种算法的平均地址分配率的最小差距接近 20%。因此, AAN 可以更加合理、高效地利用网络地址,除了受网络地址空间的限制外,并不受其他参数的限

制,可以接纳更多的节点加入网络,使其地址空间得以更充分的利用,提升了地址分配的成功率。不仅如此,AAN 还能减少孤立节点个数,增加网络连接性,有效提升地址分配成功率。

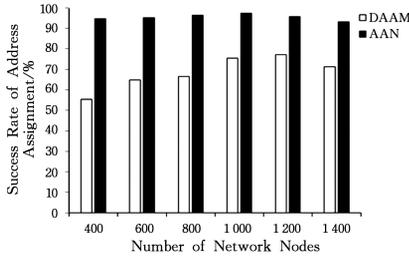


图 6 地址分配成功率

Fig. 6 Success rate of address assignment

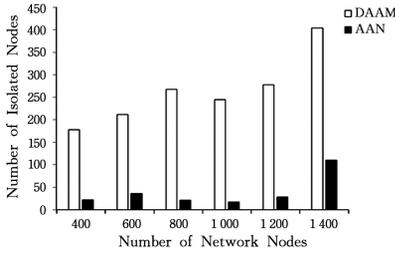


图 7 孤立节点数分析

Fig. 7 Analysis of number of isolated nodes

由图 7 可知,在参数不变的情况下,随着网络节点的增加,孤立节点数会显著增加。但在相同的网络环境下,AAN 中的孤立节点数量大大少于 DAAM 中的孤立节点数量,并且 AAN 能够有效地减少孤立节点的数量,增加节点地址分配成功率,提高网络性能。

### 5.2.2 网络最大深度

从图 8 中可以看出,DAAM 在 5 个不同场景中的网络最大深度明显比 AAN 低。主要原因在于,DAAM 在网络建立之前已经给定了最大网络深度  $L_{max}$  参数的限制,一旦网络达到  $L_{max}$  时,该网络深度将不再增加。其优点是可以有效控制网络的最大深度,进而控制传输因跳数增加所造成的延迟。但这将严重影响网络的地址分配成功率,导致距离协调器节点较远的其他节点无法加入网络。而 AAN 不受网络最大深度的限制,可以随着节点的入网不断地增加网络深度,因此最终形成的网络的最大深度较深。与此同时,定义  $D$  为网络节点的深度,不同网络深度下已入网的节点数与入网节点总数的比例为  $R = \frac{N_D}{N}$ ,其中  $N_D$  表示深度为  $D$  的已入网的节点数。从表 2 可以看出,节点刚开始分配地址时,DAAM 的节点入网率略高于 AAN,但是随着网络深度的增加,在相同深度的情况下 AAN 的节点入网速率将明显高于 DAAM。当  $D=7$  时,AAN 已经完成了入网节点总数的 90% 以上的节点地址分配任务,而 DAAM 只能做到 80%。

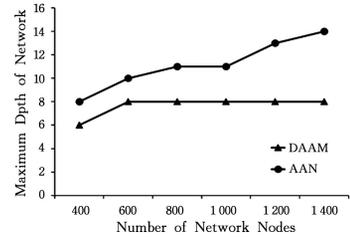


图 8 网络的最大深度

Fig. 8 Maximum depth of network

表 2 不同网络深度下入网节点的分布状况

Table 2 Distribution of nodes in different network depths

(单位:%)

	N=400		N=600		N=800		N=1000		N=1200	
	DAAM	AAN	DAAM	AAN	DAAM	AAN	DAAM	AAN	DAAM	AAN
$D=0$	0.43	0.29	0.24	0.18	0.18	0.13	0.13	0.11	0.11	0.09
$D=1$	2.58	1.73	1.67	1.45	1.27	1.03	1.05	0.81	0.85	0.68
$D=2$	4.72	4.90	2.87	4.54	4.36	3.48	3.68	3.66	2.77	3.32
$D=3$	6.44	13.26	6.01	10.71	9.07	7.75	7.37	14.02	8.62	8.16
$D=4$	9.01	15.56	11.05	13.97	12.52	14.46	11.45	20.01	11.27	15.64
$D=5$	13.73	22.48	22.16	17.79	18.87	19.51	17.78	25.90	20.11	24.32
$D=6$	23.18	30.54	26.20	23.41	27.59	25.57	23.84	27.23	28.08	23.98
$D=7$	23.60	6.63	19.46	13.07	14.34	14.33	20.28	5.49	17.13	15.05
$D=8$	16.31	4.61	10.34	13.43	11.80	9.16	14.35	1.22	11.06	4.86
$D \geq 9$	0.00	0.00	0.00	1.45	0.00	4.58	0.00	1.55	0.00	3.90
$S = \frac{N_S}{N}$	58.15	86.75	69.33	91.83	68.87	96.75	75.92	98.30	78.32	97.89

### 5.2.3 $R_{max}$ 对网络地址分配的影响

固定仿真实验中的网络节点数为 600,最大网络深度  $L_{max}$  取 8,在其他参数不变的情况下,观察不同  $R_{max}$  值对网络地址分配的影响。

从图 9 中可以明显地看出,随着  $R_{max}$  的增加,两种算法的地址分配成功率都在不断地增加,这主要是因为网络节点数与  $R_{max}$  呈现正相关的关系, $R_{max}$  越大,网络中可入网的路由节点数越多,整个网络中可入网的节点数也就越多,孤立节点数就越少。

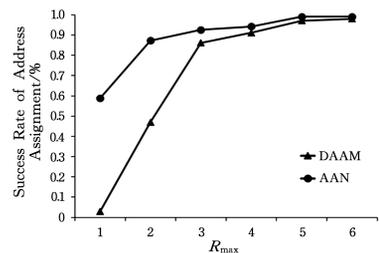


图 9 不同  $R_{max}$  下的地址分配成功率

Fig. 9 Success rate of address assignment under different  $R_{max}$

从图 10 中可以发现,在 AAN 中,网络最大深度与  $R_{\max}$  呈现负相关,随着  $R_{\max}$  值的增加,网络深度会不断地减小;而由于  $L_{\max}$  参数的限制,DAAM 中网络的最大深度与  $R_{\max}$  无关。DAAM 受网络深度参数  $L_{\max}$  的限制,无法在  $L_{\max}$  跳以内加入网络的节点都将会成为孤立节点,从而导致产生大量的孤立节点,降低了地址利用率。

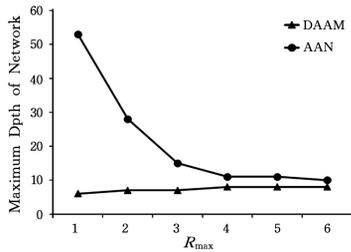


图 10 不同  $R_{\max}$  对网络最大深度的影响

Fig. 10 Effect of maximum depth of network under different  $R_{\max}$

**结束语** 本文基于 IEEE 802.15.4 规范,针对 ZigBee 技术 DAAM 算法中闲置地址无法充分利用的问题,提出了一种新的树型网络地址分配算法(AAN)。该算法避免了在网络建立过程中产生大量的孤立节点,使得地址空间的分配更加灵活。在无线随意网络(Wireless Ad hoc Network)的应用中,由于节点加入的不确定性,如何尽可能地确保大多数的节点加入网络是一个重要的研究课题。我们可以使用一种集中式的排程演算法 Top-Down-Maximum-Load-First(TDMLF)来解决隐藏节点问题<sup>[19]</sup>,但是本文并未探究网络在后续运行中的能量损耗问题,后续可以依照网络的不同应用来设计和改进不同方案以解决该问题。同时,相较于传统的路由方法,本文提出的 AAN 路由方式不再需要通过路由表进行信息传输,可以快速地进行路由,减少路由表的建立与维护所需的时间与储存空间。仿真实验结果表明,所提算法在地址分配成功率 and 孤立节点数等方面优于传统的 DAAM。

## 参考文献

- [1] ZigBee Specification Version 2008. ZigBee Document 053474-r17[S]. 2008.
- [2] ZHAN C, ZENG Y, ZHANG R. Energy-efficient data collection in UAV enabled wireless sensor network[J]. IEEE Wireless Communications Letters, 2018, 7(3): 328-331.
- [3] AL-HARBAWI M, RASID M F A, NOORDIN N K. Improved tree routing (ImpTR) protocol for ZigBee network[J]. International Journal of Computer Science and Network Security, 2009, 9(10): 146-152.
- [4] LIN S Z, LIU J Y, FANG Y J. ZigBee based wireless sensor networks and its applications in industrial[C]// Proceedings of the IEEE International Conference on Automation and Logistics. IEEE, 2007: 1979-1983.
- [5] DOU Z, WANG X, LI Y. Addressing for 6LoWPAN[J]. International Journal of Internet Protocol Technology, 2019, 12(1): 51-60.
- [6] ZHOU Y, FADLULLAH Z M, MAO B. A Deep-Learning-Based Radio Resource Assignment Technique for 5G Ultra Dense Networks[J]. IEEE Network, 2018, 32(6): 28-34.
- [7] DU Z G, QIAN D P, LIU Y. Address Allocation Protocol in Wireless Sensor Networks [J]. Journal of Software, 2009, 20(10): 2787-2798.
- [8] PAN M S, TSAI C H, TSENG Y C. The Orphan Problem in ZigBee Wireless Networks[J]. IEEE Transactions on Mobile Computing, 2009, 8(11): 1573-1584.
- [9] GIRI D, ROY U K. Address borrowing in wireless personal area network[C]// Proceedings of the IEEE International Advance Computing Conference. IEEE, 2009: 181-186.
- [10] FANG M Q, WAN J, XU X H. A preemptive distributed address assignment mechanism for wireless sensor networks[C]// Proceedings of the International Conference on Wireless Communications. IEEE, 2008: 1-5.
- [11] LI Y R, SHI H B, TANG B Y. Address assignment and routing protocol for large-scale uneven wireless sensor networks[C]// Proceedings of the International Symposium on Computer Network and Multimedia Technology. IEEE, 2009: 1-4.
- [12] GAMI S, KRISHAKUMAR A S, KRISHNAN P. Infrastructure-based location estimation in WLAN networks[C]// Proceedings of the IEEE Wireless Communications and Networking Conference. IEEE, 2004, 1: 465-470.
- [13] LORIINCZ K, WELSH M. Motetrack: A robust, decentralized approach to RF-based location tracking[C]// Proceedings of the International Symposium on Location and Context-Awareness. 2005: 63-82.
- [14] REN Z, LI P X, YAO Y K, et al. Segmentation-based Extensible Address Assignment Algorithms for ZigBee Networks on Demand[J]. Journal of Communications, 2012, 33(5): 131-137.
- [15] PERES B S, SANTOS B P, OTAVIO A O. Matrix: Multihop address allocation and dynamic any-To-any routing for 6LoWPAN [J]. Computer Networks, 2018, 140: 28-40.
- [16] YEN L H, TSAI W T. Flexible address configurations for tree-based ZigBee/IEEE 802.15.4 wireless networks[C]// Proceedings of the IEEE International Conference on Advanced Information Networking and Applications. IEEE, 2008: 395-492.
- [17] LI X H, FANG K L, GU J G. An improved ZigBee routing strategy for monitoring system[C]// Proceedings of the IEEE International Conference on Intelligent Networks and Intelligent Systems. IEEE, 2008: 255-258.
- [18] ULLAH I, YOUN H Y. Statistical Multipath Queue-Wise Preemption Routing for ZigBee-Based WSN[J]. Wireless Personal Communications, 2018(6): 1-15.
- [19] WANG X X. The Bottleneck Problem in IEEE 802.15.4 / ZigBee Cluster-Tree Networks[D]. Taiwan: National Taiwan Ocean University, 2010.



**LIU Ning-ning**, born in 1993, postgraduate, is member of China Computer Federation (CCF). His main research interests include parallel and distributed computing, wireless sensor networks and so on.



**LIN Cheng-kuan**, born in 1976, professor, is member of China Computer Federation (CCF). His main research interests include graph theory, design and analysis of algorithms, discrete mathematics, wireless sensor networks, mobile computing and parallel and distributed computing.