

基于分布式邻居发现机制的 ZigBee 网络能耗研究

黄恒杰¹ 周涛² 王高才²

(玉林师范学院教育技术中心 玉林 537000)¹ (广西大学计算机与电子信息学院 南宁 530004)²

摘要 ZigBee 是一种基于 IEEE 802.15.4 标准的适用于短距离、低功耗和低数据传输率的无线网络技术。ZigBee 网络的节点电池能量的有效利用能延长其网络生命周期。采用分布式邻居发现机制研究了 ZigBee 网络能耗。该机制使得 ZigBee 网络中的节点在其可访问的区域内及时发现邻居节点,通过分析和定义 3 种分布式邻居发现算法下 ZigBee 节点在整个发现过程中所需要的平均帧数,提出了 ZigBee 网络能耗模型,对 ZigBee 网络设备在 3 种分布式邻居发现算法下的能耗进行了详细分析并得出能耗表达式。在仿真实验中比较了 ZigBee 网络节点和邻居节点在不同算法下的工作循环模式和连续模式下设备的能耗,结果表明对于基于分布式邻居发现机制算法的 ZigBee 网络能耗,当帧时隙数较小时,选择争用树算法能获得较好的节能效果;当帧时隙数较大时,选择无反馈帧时隙算法或有反馈帧时隙算法则可减少节点能耗以延长 ZigBee 网络的生命周期。

关键词 ZigBee 网络,分布式邻居发现,平均帧数,能耗模型

中图分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.5.012

Study on Energy Consumption of ZigBee Networks Based on Distributed Neighbor Discovery Mechanism

HUANG Heng-jie¹ ZHOU Tao² WANG Gao-cai²

(Educational Technology Center, Yulin Normal University, Yulin 537000, China)¹

(School of Computer and Electronic Information, Guangxi University, Nanning 530004, China)²

Abstract ZigBee networks are short-distance, low-energy consumption and low-data transmission rate wireless network technology based on IEEE 802.15.4 standard. Usually, its network life can be extended if the battery energy is utilized with high efficiency. In this paper, we mainly focused on ZigBee networks energy consumption based on distributed neighbor discovery mechanism. The mechanism can make the node of ZigBee find other neighbor nodes in an accessible area. The paper analyzed and obtained the number of required average frame in the discovering process for three distributed neighbor discovery algorithms, and presented energy consumption model for ZigBee networks. The energy consumptions of the requesting device and neighbor devices in these algorithms were analyzed and energy consumption expressions were given respectively. In simulation, we compared the energy consumption of the duty cycle and continuous mode of devices. The results show that the ZigBee network energy consumption, which is based on the distributed neighbor discovery algorithm, can achieve better energy saving effect when the time slot number is small if the competitive tree algorithm(CTA) is selected. And when the number of frame time slot is large, the framed slotted ALHOA without feedback packet(FSA_noFBP) and framed slotted ALHOA with feedback packet(FSA_FBP) ought to be selected.

Keywords ZigBee networks, Distributed neighbor discovery, Number of average frame, Energy consumption

1 引言

ZigBee 是一种新兴的基于 IEEE 802.15.4 标准的双向通信无线网络技术,它主要工作在 3 个频带:2.4GHz、915MHz 和 868MHz。ZigBee 技术具有近距离、低功耗、低成本、低速率、高可靠性和组网灵活等特点。其由于网络容量大、时延

短、数据安全性较高,因此被广泛应用于多种领域。ZigBee 网络包含由 IEEE 802.15.4 标准定义两种功能类型的物理层设备:全功能设备(Full Function Device, FFD)和精简功能设备(Reduced Function Device, RFD)。在 ZigBee 网络中,所有的节点可分为 3 种基本类型:网络协调器节点、路由节点以及终端节点。全功能设备 FFD 可做为节点充当网络中的协

到稿日期:2015-07-13 返修日期:2015-11-28 本文受国家自然科学基金:基于可调度性的网络演算的绿色网络能耗优化研究(61262003),移动环境下基于博弈论的能量感知协同内容分发策略研究(61562006),广西自然科学基金杰出青年基金(2013GXNSFGA019006)资助。

黄恒杰(1980—),硕士,工程师,主要研究领域为网络优化,E-mail:4854385@qq.com;周涛(1987—),硕士生,主要研究领域为无线网络技术,E-mail:1006372922@qq.com;王高才(1976—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机网络技术、系统性能评价和随机方法,E-mail:gcwang@gxu.edu.cn。

调器和路由器,所以一个网络至少含有一个 FFD;精简功能设备 RFD 只能与主节点进行通信,它的实现相对简单,所以只作为终端设备节点。路由节点是网络中的无线收发器,所以必须是全功能设备。由此,ZigBee 网络将两种物理设备定义成 3 种逻辑设备类型:ZigBee 协调器、ZigBee 路由器和 ZigBee 终端设备。

(1) ZigBee 网络协调器:协调器是启动和配置网络的设备,它主要负责构建、管理和维护网络,维护间接寻址用的绑定关系表,完成成员身份和链路状态信息管理。相对于其他设备,它拥有更大的存储容量和计算能力。一个 ZigBee 网络只能有一个唯一的协调器。

(2) ZigBee 网络路由器:其主要功能是扩展网络以及路由消息。网络路由器作为 ZigBee 网络的潜在父节点时,它允许更多的节点设备接入该网络中。ZigBee 网状网络和树状网络可以拥有多个 ZigBee 路由器,而 ZigBee 星型网络不支持路由器功能,不存在路由节点。

(3) ZigBee 终端设备:终端设备不具备路由和网络协调能力,只作为 ZigBee 网络中的终端节点,主要负责采集监控对象的信息。它只与其父节点主动通信,因为它不具有路由功能,所以必须将消息传输给具有路由能力的父节点以及网络中的协调器节点。终端设备的存储器容量要求最小。

ZigBee 网络通常在无人值守的状态下工作,要求网络中节点的平均能耗比现有无线网络(如 Bluetooth)中的能耗更低。在一些工业监控应用中,装备纽扣电池的 ZigBee 网络需要在无人值守的情况下工作几个月甚至几年。在森林火灾检测等大范围的环境监测应用中,为大量的 ZigBee 网络节点频繁地更换电池是不现实的;在矿井人员定位应用中,为每个 ZigBee 网络节点提供长期电源也是不现实的。因此,ZigBee 节点的生命周期依赖于有限的电源。这些典型的应用要求 ZigBee 网络在运行过程中,每个节点都要最小化自身的能量消耗,使得 ZigBee 网络获得最长的生命周期。所以 ZigBee 网络的生命周期是一个非常关键的参数。ZigBee 的节能分层为:在物理层(Physical Layer, PHY)降低硬件的能耗,在媒体接入层(Media Access Layer, MAC)对能量进行动态的管控(如收发器的工作模式切换),在网络层中优化路由机制和数据传输控制,从而延长整个 ZigBee 网络的生命周期,以达到节能的目的。

目前,无线通信系统的终端(如 ZigBee 终端)电池寿命通常是 2~3 天,其高能耗主要是因为终端在空闲状态下需要周期性接收系统广播信道,在激活状态下需要周期性接收公共控制信道,睡眠时间短,需要支持自适应操作、切换和移动性管理大量的测量、反馈、信令等。因此本文主要利用分布式邻居发现机制算法优化 ZigBee 网络的能耗。分布式邻居发现机制算法使 ZigBee 网络中的一个 ZigBee 终端在其可访问的区域里发现和识别能够提供连接的 ZigBee 网络路由的邻居节点。与目前的相关节能机制相比,所提方法更为具体,如当前集中于从跨层角度考虑优化和改进能耗,或引入某种节点能耗均衡策略。本文第 2 节给出了 ZigBee 网络和相关分布式邻居发现机制算法理论的介绍;第 3 节对系统模型和分布式邻居发现机制算法平均帧数求解;第 4 节详细分析并给出了不同算法下 ZigBee 网络能耗的表达式;第 5 节给出了实验及分析结果;最后对本文的研究工作进行总结。

2 相关研究工作

当前,许多研究工作从性能和能耗两方面分析 ZigBee 网络。文献[1]针对 ZigBee 网络的能耗优化机制进行了研究与验证,分别从网络休眠机制、路由机制、协议栈具体实现 3 个方面进行了优化与改进。文献[2]提出一种 ZigBee 网络中基于跨层机制的能耗均衡路由算法,构建网关至节点的下行多径路由,引入跨层机制更新邻居节点的剩余能量信息等。文献[3]给出了 ZigBee 节点能耗的计算方法。对于不同的邻居发现机制,文献[4]研究了帧时隙算法,节点标识的每个请求时间帧被分成很多个时隙。为了发送其标识,每个标签随机选择一个帧时隙,并不断重复,直到解码所有标签的标识。帧的大小取决于具体实现算法^[5],可以是固定或可变的。固定帧大小的帧时隙(Framed Slotted ALHOA, FSA)算法称为基本的帧时隙(Basic Framed Slotted ALHOA, BFSa)算法;可变帧大小的 FSA 称为动态的帧时隙(Dynamic Framed Slotted ALHOA, DFSA)算法^[6,7]。BFSa 和 DFSA 可以根据它们是否支持静音和早期结束进行进一步划分。静音是使阅读器(Reader)对已经被成功读取的标签不响应。而早期结束是使阅读器提前关闭空闲时隙或无响应时隙。文献[8]中提出动态帧时隙方法,使用标签评估方式评估阅读器周围的标签数,根据标签数动态分配帧大小,结果表明提出的动态帧时隙方法具有更低的复杂性和较好的延时性能。文献[9,10]提出用于评估给定数目的标签在一个帧内竞争选择最佳帧长度并用于后续帧的算法。文献[11]考虑被识别的标签从接下来的帧竞争中退出,评估识别一组标签所需要的平均帧数,结果表明对于给定的帧长度,用于竞争的帧数随着标签数的增加呈指数增加。

在争用树算法(Competitive Tree Algorithm, CTA)中^[12],识别过程由包含多个时隙的一组帧组成,为了减少冲突数目,将标签分成不同的子集。在第一个帧里,每个标签随机选择一个时隙来发送它的标识。因此,当两个或者两个以上的标签选择相同的时隙时,冲突发生。此时,需分配一个新帧重新发送标签以解决冲突。于是,一个帧中只要有一个时隙冲突,就要分配一个新帧。当阅读器能够从所有的标签中成功接收标识包时,算法结束,即最后帧不存在时隙冲突。文献[13]的争用树算法利用三重(冲突/成功/空闲侦听)反馈。假设一棵新树有 n 个争用发送器,即 n 个发送器想广播一个数据包。在第一个争用帧期间即树根阶段, n 个发送器以相同概率随机选择 1 到 m 之间的任意争用时隙 k 发送包。争用帧完成后,三重反馈可用,每个发送器知道它的包是否广播成功,如果没有,一个新的争用帧分配给在特定时隙期间引起冲突的发送器。因此,如果所有帧时隙都存在冲突, m 个新争用帧可用。这形成一棵度为 m 的树。树扩展为空或者时隙成功时停止,则 n 个争用者成功地广播它们的数据,然后新树算法再次开始。争用树为多访问广播通信信道提供高效方案,定义提供给所有争用者访问所需要的树层数和帧数。文献[14]使用争用树算法研究标签耗,提出竞争树算法的能量分析,但其没有考虑阅读器的能耗,结果表明当帧长度增加时标签的能耗减少,且当标签数不断增加时能耗稳定于一个恒定值。

基于二分法树的 RFID 标签协议^[15]源于最初的多访问信道协议,该方法能够解决所有标签冲突。在该协议下一个

周期的一帧里,如果冲突发生,引起冲突的标签被分成两个子项,且这些标签独自决定加入哪个子项。第一个子项的标签直接将它们的识别发送到下一帧,当第一个子项的所有识别的标签被收集后,第二个子项的标签再将它们的识别发送到下一帧。这个“分而治之”的过程重复且递归地执行,直到最后的子项至多有一个标签,此时收集标签的识别不发生冲突。基于二进制查询树的协议通过广播一些特定设计的二进制串以执行子项部分。假设在一个循环 i 帧中,阅读器广播的二进制串是 S_i (对于在一个周期的首帧),在阅读器范围的每个标签通过解码二进制串,比较它的二进制标识是否从 S_i 开始。如果是,则标签调整数据识别到信号且在帧的数据隙里发送波形,否则标签保持沉默且在帧里不响应。如果在数据隙里发生冲突,这些协议在 S_i 后添加 0。阅读器广播 $S_{i+1} = S_i 0$ 到下一帧,且收集二进制识别以 $S_i 0$ 开始的标签。当这些标签的识别被收集之后,二进制查询树协议将 1 添加到 S_i 后,然后阅读器收集以 $S_i 1$ 开头的二进制标识标签。这个过程不断重复,直到所有标签冲突解决,且所有标签识别被成功收集。文献[16]使用查询树算法研究标签的能耗,查询树算法是争用树算法的具体执行,在查询树算法的每个请求中,阅读器发送一个含有二进制前缀的查询。具有二进制前缀的标识标签在单个时隙返回响应,当超过一个标签响应时冲突发生,当只有一个冲突时,阅读器增加一位(0 或者 1)给前缀且发送一个新的(更长的)前缀查询,重复这个过程直到所有的标签标识被阅读器成功收集。

3 系统模型及分布式邻居发现机制算法的平均帧数求解

3.1 系统模型

分布式邻居发现类似于无线射频识别(RFID),由一个请求设备发起,以发现发送范围内的邻居设备。假设一个单跳 ZigBee 区域网络包含一个请求设备(网络协调器)和 n 个邻居设备(网络路由器或终端设备),则 ZigBee 设备具有 3 种状态:发送、接收和睡眠,对应的功率分别是 P_{tx} 、 P_{rx} 和 P_{sleep} ,并假设睡眠和活动状态(即发送或者接收状态)之间的切换功率忽略不计。

请求设备发送一个识别命令以启动发现过程,然后等待邻居设备发送 ID 包,为它分配一段固定时间,其被划分成 m 个时隙的帧。假设每个邻居设备正在侦听信道,则随机选择时隙发送 ID,若此时只有一个 ID 被发送,则 ID 发送成功;若有多个 ID 发送,则产生时隙冲突(假设设备不存在由无线信道传输带来的错误,且当 ID 包被请求设备成功解码时邻居设备能被发现)。

当邻居设备处于下述两种模式时不会发送 ID 包:

(1)工作循环模式:为了减少能耗切换到了睡眠状态;

(2)连续模式:为了接收单跳覆盖范围内的其它邻居设备的 ID,仍然在侦听无线信道。

上述发现过程在一系列帧中不断重复,直到请求设备发现 n 个邻居设备为止。请求设备在发现 n 个邻居设备的最后一帧之后发送一个反馈包(Feedback Packet, FBP)以结束发现过程。此外,请求设备在每个帧之后发送一个 FBP 以告知邻居设备每帧的成功时隙,这依赖于帧冲突的解决(冲突解决算法)。本文主要考虑 3 种分布式邻居发现算法:每帧无反馈

帧时隙(FSA_noFBP)算法、每帧有反馈帧时隙(FSA_FBP)算法和争用树算法(CTA)。其中 FSA_FBP 和 CTA 在每帧之后发送一个 FBP,而 FSA_noFBP 只需在最后一帧发送 FBP。

为了研究不同分布式邻居发现算法下 ZigBee 网络的能耗,必须求得 ZigBee 设备在该算法下的发现邻居设备所需的平均帧数,从而获得发送这些帧数所消耗的能量,下面对上述 3 种算法的平均帧数进行求解。

3.2 每帧无反馈帧时隙算法

对于 FSA_noFBP 算法,请求设备仅在最后一帧后发送一个 FBP。对于整个发现过程,首先评估发现给定邻居设备且置信水平为 α 所需的平均帧数 T_n ,从而得到空闲侦听、成功响应和产生冲突的时隙数。发现过程评估如下:设帧大小为 m ,邻居设备数为 n ,根据二项分布, k 个邻居设备选择一个公用时隙发送 ID 的概率为:

$$p_{n,m}(k) = \binom{n}{k} \left(\frac{1}{m}\right)^k \left(1 - \frac{1}{m}\right)^{n-k} \quad (1)$$

其中, $\binom{n}{k} = \frac{n!}{k!(n-k)!}$,因此,一帧的空闲时隙数 $m_{idle}(n, m)$ 、成功时隙数 $m_{succ}(n, m)$ 和冲突时隙数 $m_{coll}(n, m)$ 的数学期望分别为:

$$m_{idle}(n, m) = m \times p_{n,m}(0) = m \left(1 - \frac{1}{m}\right)^n \quad (2)$$

$$m_{succ}(n, m) = m \times p_{n,m}(1) = n \left(1 - \frac{1}{m}\right)^{n-1} \quad (3)$$

$$m_{coll}(n, m) = m - m_{idle}(n, m) - m_{succ}(n, m) \quad (4)$$

于是,一帧的邻居设备识别率 r 为:

$$r(n, m) = \frac{m_{succ}(n, m)}{n} = \left(1 - \frac{1}{m}\right)^{n-1} \quad (5)$$

在 T_n 帧后仍然没有发现邻居设备的概率是:

$$p_{miss}(i) = \prod_{i=1}^{T_n} (1-r) = 1 - \alpha \quad (6)$$

其中, α 是发现邻居设备的置信水平,由于 r 在 FSA 里对于所有发现过程相同,式(6)变为:

$$(1-r)^{T_n} = 1 - \alpha \quad (7)$$

对于 T_n ,发现置信水平为 α 的一组邻居设备所需的平均帧数至少是:

$$T_n \geq \left\lceil \frac{\log(1-\alpha)}{\log(1-r)} \right\rceil \quad (8)$$

其中, $\lceil x \rceil$ 表示不小于 x 的最小整数。

3.3 有反馈帧时隙算法

对于 FSA_FBP 算法,请求设备在每帧之后发送一个 FBP,当邻居设备被发现后不再发送新的 ID 包。因此,竞争邻居设备数不断减少,直到发现给定数量的邻居设备时停止。

对于帧大小为 m 的帧时隙法协议,假设有 n 个邻居设备在请求设备附近,且在 i 帧之后还未发现的邻居设备数为 n_i ,其中 $n_0 = n$,则在 $i+1$ 帧时发现的邻居设备数为 $n_i \left(1 - \frac{1}{m}\right)^{n_i-1}$,而在 $i+1$ 帧后尚未发现的邻居设备数 n_{i+1} 表示为:

$$n_{i+1} = n_i - n_i \left(1 - \frac{1}{m}\right)^{n_i-1} \quad (9)$$

因此,第 $i+1$ 帧中邻居设备的识别率 r_{i+1} 为:

$$r_{i+1}(n_{i+1}, m) = \frac{m_{succ}(n_{i+1}, m)}{n_{i+1}} = \left(1 - \frac{1}{m}\right)^{n_{i+1}-1} \quad (10)$$

这个发现过程不断重复,直到所有邻居设备被发现时停

止。最后,将式(10)代入式(8)得到 FSA_FBP 算法所需的平均帧数 T_n 。

3.4 争用树算法

CTA 的竞争解决过程如图 1 所示,树的每个节点代表由 m 个时隙组成的一帧,请求设备在每帧之后发送一个 FBP。在第一帧(即树的根),每个邻居设备随机从 m 个时隙中选择一个发送它的 ID 包。如果邻居设备在选择的时隙里发送 ID 包不成功,则一个新的帧被分配给该设备。因此,如果形成 k 个时隙冲突,那么 k 个新帧在当前帧之后可用,从而形成一棵树,这棵树扩展到帧上即树的叶子节点,仅仅包含空的或者成功的时隙,即不存在冲突时隙。

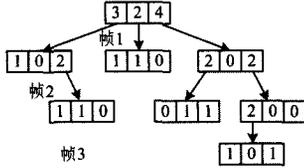


图 1 争用树算法实现示意图

例如,图 1 描述了 $n=12$ 时邻居设备 CTA 的实现示意图,每帧有 $m=3$ 个时隙。每个时隙的数字表示在该时隙中发送 ID 包的设备数。如果这个数字大于 1,即有冲突,那么冲突设备的新帧被添加到树的下一层;如果这个数字等于 0 或 1,即空闲或成功,那么表明该帧的时隙不被设备占用或只被一个设备占用。

图 1 的前 3 帧代表争用树算法的实现过程:帧 1 是树的根,在帧 1 的第一个时隙里有 3 个设备竞争,这 3 个设备被分配到帧 2,其中有 2 个设备在帧 2 的第 3 个时隙里竞争,这 2 个设备被分配到帧 3,而帧 3 不存在竞争。

下面分析竞争树算法的概率,目标是发现所有可访问的邻居设备。对于一个需要成功发送 ID 包的给定邻居设备,首先定义树的层数 l_n ,然后完成竞争树算法所需的平均帧数 \bar{F}_n 。图 1 的邻居设备成功发送 ID 包所需要的层数是 3、3 和 4,这里有 $n=12$ 个邻居设备,平均层数 $\bar{l}_{12}=35/12$,且总共帧数是 $F_{12}=9$ 。

接下来使用 l_n 和 \bar{F}_n 分别表示基于争用树算法的分布式邻居发现的层数和帧数。

(1) CTA 的平均层数 l_n

当邻居设备数为 n 且每帧时隙数为 m 时,一个邻居设备成功发送它的 ID 包所需要的层数 l (l 是一个邻居设备竞争直到它成功为止的树的层数)的概率分布如下:

$$P(l|n) = \begin{cases} 0, & n=1, l \neq 1 \\ 1, & n=1, l=1 \\ p_s(l) - p_s(l-1), & n \geq 2 \end{cases} \quad (11)$$

其中, $p_s(l)$ 是一个邻居设备在 l 层成功的概率,即选择 l 层的一个不被其他邻居设备占用的时隙的概率,且 $p_s(l)$ 与 $p_s(l-1)$ 的偏差等于邻居设备确定需要单独占用 l 层的争用时隙的概率。由于每帧时隙数为 m ,在 l 层总共的时隙数为 m^l ,根据式(5), $p_s(l)$ 表示为:

$$p_s(l) = (1 - \frac{1}{m^l})^{n-1} \quad (12)$$

其中, $\frac{1}{m^l}$ 是邻居设备选择 l 层的一个时隙的概率。通过概率分布可以计算平均层数 \bar{l}_n 为:

$$\bar{l}_n = \sum_{l=1}^{\infty} l P(l|n) \quad (13)$$

由文献[14]可得:

$$\bar{l}_n = \log_m(n-1) + (\frac{1}{2} + \frac{\gamma}{\log m}) + \frac{1}{2n \log m} \quad (14)$$

其中,欧拉常量 $\gamma \approx 0.5772$ 。对于一个给定的 m , \bar{l}_n 随着 n 对数级增加。

(2) CTA 的平均帧数 \bar{F}_n

定理 对于 n 个邻居设备且每帧有 m 个时隙的情况,竞争树算法所需的平均帧数为 \bar{F}_n ,其可递归地定义为:

$$\bar{F}_n = \begin{cases} 1, & n=1 \\ \frac{m}{m-1}, & n=2 \\ (1 - m^{1-n})^{-1} [1 + \sum_{k=2}^{n-1} \binom{n}{k} \frac{(m-1)^{n-k}}{m^{n-1}} \bar{F}_k], & n \geq 3 \end{cases} \quad (15)$$

证明:(1)当 $n=1$ 时,显然 $\bar{F}_1=1$;

(2)当 $n=2$ 时,

$$\begin{aligned} \bar{F}_2 &= 1 + m \cdot P\{N_i=2|N=n\} \cdot \bar{F}_2 \\ &= 1 + m \cdot P\{N_i=2|N=2\} \cdot \bar{F}_2 \\ &= 1 + m \cdot \binom{2}{2} \left(\frac{1}{m}\right)^2 (1 - \frac{1}{m})^0 \cdot \bar{F}_2 \\ &= 1 + \frac{1}{m} \cdot \bar{F}_2 \end{aligned} \quad (16)$$

从而可得: $\bar{F}_2 = \frac{m}{m-1}$;

当 $n \geq 3$ 时,首先定义 \bar{F}_k' :

$$\bar{F}_k' = \begin{cases} 0, & k=0 \\ \bar{F}_k, & k>0 \end{cases} \quad (17)$$

然后, \bar{F}_n 表示为:

$$\begin{aligned} \bar{F}_n &= 1 + \sum_{i=1}^m \sum_{k=0}^n P\{N_i=k|N=n\} \cdot \bar{F}_k' \\ &= 1 + m \sum_{k=0}^n P\{N_i=k|N=n\} \cdot \bar{F}_k' \\ &= 1 + m \sum_{k=1}^n P\{N_i=k|N=n\} \cdot \bar{F}_k \end{aligned} \quad (18)$$

根据二项分布概率可得:

$$\begin{aligned} P\{N_i=k|N=n\} &= \binom{n}{k} \left(\frac{1}{m}\right)^k (1 - \frac{1}{m})^{n-k} \\ &= \binom{n}{k} \frac{(m-1)^{n-k}}{m^n} \end{aligned} \quad (19)$$

将式(19)代入式(18)可得:

$$\begin{aligned} \bar{F}_n &= 1 + m \sum_{k=1}^n \binom{n}{k} \frac{(m-1)^{n-k}}{m^n} \cdot \bar{F}_k' \\ &= 1 + m^{1-n} \sum_{k=1}^n \binom{n}{k} (m-1)^{n-k} \cdot \bar{F}_k + m^{1-n} \bar{F}_n \end{aligned} \quad (20)$$

整理可得式(15)。

证毕。

然而,根据文献[14]有:

$$\bar{F}_n = 1 + \sum_{k=1}^{\infty} c_n(m^k) \quad (21)$$

其中, $c_n(m^k)$ 是 k 层有冲突的竞争时隙平均数,即平均帧数在下一层 $k+1$ 得出,根据式(4)可得:

$$c_n(M) = M - M(1-1/M)^n - n(1-1/M)^{n-1} \quad (22)$$

其中, $M=m^k$,且 $M(1-1/M)^n$ 和 $n(1-1/M)^{n-1}$ 分别表示 k

层的空闲时隙数和成功时隙数。根据傅里叶分析,平均帧数 $\overline{F_n}$ 可表示为:

$$\overline{F_n} = \frac{n}{\log m} - \frac{1}{m-1} \quad (23)$$

可以看出,对于给定的 m , 平均帧数 $\overline{F_n}$ 随 n 线性增加。

4 网络能耗模型描述

在 ZigBee 网络中,请求设备的能耗由发现给定邻居设备的持续时间决定。如果 D 是发现 n 个邻居设备的总延时,那么发现过程中一个请求设备的能耗为:

$$E = P \times D \quad (24)$$

其中, E (焦耳) 是请求设备发现 n 个邻居设备的能耗, P (瓦特) 是请求设备在发现过程中的功率, D 是发现过程的持续时间。

4.1 请求设备的能耗分析

请求设备的平均能耗用 $\overline{E_R}$ 表示。下面给出不同邻居发现算法的 $\overline{E_R}$ 。

(1) 每帧无反馈帧时隙算法

使用每帧无反馈的帧时隙算法时,请求设备在 T_n 个帧里重复执行如下的操作:在帧的 m 个时隙侦听信道以接收进入的 ID 包;在最后帧之后发送一个 FBP 来结束发现过程。因此使用每帧无反馈的帧时隙算法请求设备的平均能耗 $\overline{E_R(FSA_{noFBP})}$ 为:

$$\overline{E_R(FSA_{noFBP})} = T_n \cdot m P_{rx} T_{SLOT} + P_{rx} T_{FBP} \quad (25)$$

其中, m , T_n , T_{SLOT} , T_{FBP} , P_{rx} 和 P_{rx} 分别表示完成发现所需的每帧时隙数、平均帧数、一个时隙的持续时间、反馈包的持续时间、发送状态的功率和接收状态的功率。

(2) 每帧有反馈帧时隙算法

对于每帧有反馈的帧时隙算法,请求设备重复执行如下操作:在每个帧的 m 个时隙侦听信道以接收进入的 ID 包;在每帧之后发送一个 FBP。因此使用每帧有反馈的帧时隙算法请求设备的平均能耗 $\overline{E_R(FSA_{FBP})}$ 为:

$$\overline{E_R(FSA_{FBP})} = T_n (m P_{rx} T_{SLOT} + P_{rx} T_{FBP}) \quad (26)$$

(3) 争用树算法

对于争用树算法,请求设备执行的操作和每帧有反馈的帧时隙法的操作一样。因此使用争用树算法请求设备的平均能耗 $\overline{E_R(CTA)}$ 为:

$$\overline{E_R(CTA)} = \overline{F_n} (m P_{rx} T_{SLOT} + P_{rx} T_{FBP}) \quad (27)$$

其中, $\overline{F_n}$ 是完成整个过程所需的平均帧数。

4.2 邻居设备的能耗分析

对于邻居设备的能耗分析,每个邻居设备的平均能耗用 $\overline{E_N}$ 表示。因此,所有邻居设备的平均总能耗是 $\overline{E_A} = n \cdot \overline{E_N}$ 。其中, $\overline{E_A}$ 是邻居设备处于工作循环模式时的平均总能耗,即它们在不发送 ID 包的时隙切换到睡眠状态。在连续模式中,用 P_{rx} 代替 P_{sleep} 来计算邻居设备的能耗。

(1) 每帧无反馈帧时隙法

在每帧无反馈的帧时隙法中,一个邻居设备在 T_n 帧里重复执行如下操作:从 m 个时隙中随机选择一个发送 ID 包;然后在 $m-1$ 个时隙进入睡眠状态;在最后帧之后接收一个 FBP。因此,使用每帧无反馈帧时隙法时,邻居设备的平均能耗 $\overline{E_N(FSA_{noFBP})}$ 可以写成:

$$\overline{E_N(FSA_{noFBP})} = T_n [P_{rx} T_{SLOT} + (m-1) P_{sleep} T_{SLOT}] + P_{rx} T_{FBP} \quad (28)$$

其中, P_{sleep} 表示设备在睡眠状态下的功率。那么所有邻居设备的平均总能耗 $\overline{E_A(FSA_{noFBP})}$ 可写为:

$$\overline{E_A(FSA_{noFBP})} = n \overline{E_N(FSA_{noFBP})} \quad (29)$$

(2) 每帧有反馈帧时隙法

使用每帧有反馈帧时隙法时,邻居设备的能耗包括已经被请求设备发现的能耗和还没被发现的能耗。

在每帧有反馈的帧时隙法和争用树算法中,邻居设备数不断减少,因为被发现的设备将不在后续帧竞争中。一个邻居设备还没被发现之前重复执行如下操作:从 m 个时隙中随机选择一个发送 ID 包;然后在 $m-1$ 个时隙进入睡眠;在每帧之后接收一个 FBP,直到被发现时停止。因此,一个邻居设备在一个帧中竞争的能耗 E_{tx_ID} 为:

$$E_{tx_ID} = P_{tx} T_{SLOT} + (m-1) P_{sleep} T_{SLOT} + P_{rx} T_{FBP} \quad (30)$$

当邻居设备被发现后,它将在接下来的每个帧中重复执行如下操作:在每帧的 m 个时隙进入睡眠;在每帧之后接收一个 FBP,直到发现过程结束。因此,一个邻居设备在一个帧中没有竞争的能耗 $E_{no_tx_ID}$ 为:

$$E_{no_tx_ID} = m P_{sleep} T_{SLOT} + P_{rx} T_{FBP} \quad (31)$$

因此,使用每帧有反馈的帧时隙法的所有邻居设备的平均总能耗 $\overline{E_A(FSA_{FBP})}$ 为:

$$\overline{E_A(FSA_{FBP})} = T_n [(n-i) E_{tx_ID} + i E_{no_tx_ID}] \quad (32)$$

其中, i 是在给定时间被发现的设备数,即它们不需要竞争, $n-i$ 是尚未被发现的设备数(从式(9)得到剩余未发现的设备数),即它们需要竞争,且 T_n 是发现 i 个邻居设备所需的平均帧数, $i = \{0, 1, \dots, n\}$, $(n-i) E_{tx_ID} + i E_{no_tx_ID}$ 表示一帧中所有邻居设备的总能耗。

(3) 争用树算法

类似每帧有反馈帧时隙法,争用树算法的所有邻居设备的平均总能耗 $\overline{E_A(CTA)}$ 为:

$$\overline{E_A(CTA)} = n [\overline{l_n} E_{tx_ID} + (\overline{F_n} - \overline{l_n}) E_{no_tx_ID}] \quad (33)$$

其中, $\overline{l_n}$ 是邻居设备竞争的平均帧数,即用于发送 ID 包的帧数,且 $\overline{F_n} - \overline{l_n}$ 是邻居设备不需要竞争的平均帧数,即它在 $\overline{l_n}$ 帧中已经成功发送了它的 ID 包, $\overline{l_n} E_{tx_ID} + (\overline{F_n} - \overline{l_n}) E_{no_tx_ID}$ 表示一个邻居设备在所有帧的能耗。

5 实验结果及分析

对于不同的帧长度(即每帧时隙数 m),无反馈的帧时隙法、有反馈的帧时隙法和争用树算法完成整个发现过程需要的平均帧数、请求设备的平均能耗和邻居设备的平均总能耗可通过 MATLAB 仿真。表 1 给出了评估 3 种分布式邻居发现算法在能耗方面的性能参数,其是依据 IEEE 802.15.4 标准和 M2M 设备 CC2520 无线电收发机的规格设置的。

表 1 实验参数设置

参数	值	参数	值
MAC 头部	9B	数据率	250kbps
ID 负载	2B	FBP 负载	$m \cdot 2 \text{bits/slot}$
CRC	2B	T_{SLOT}	544us
TFBP	$8 \cdot (\text{FBP length})$	P_{tx}	100.8mW
P_{rx}	66.9mW	P_{sleep}	525uW
α	0.99	—	—

(1) 请求设备的平均帧数

完成发现过程所需要的平均帧数用 \overline{AF} 来表示,图 2 是对于不同算法且考虑两种不同 n 值($n=10$ 或 $n=30$)的平均帧数随着帧长度增加而减小的曲线。帧时隙算法 \overline{AF} 的表达式对应式(8)中的 T_n , 争用树算法对应式(23)中的 \overline{F}_n 。

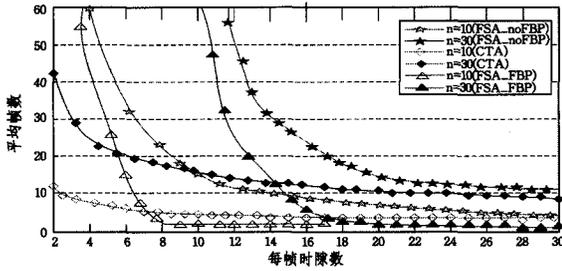


图 2 发现过程所需要的平均帧数

如上文分析,当帧长度增加时, \overline{AF} 呈指数规律递减且趋近于一个恒定的值。事实上,帧的长度越长,在一个时隙里冲突的概率就越低,那么每帧发现的设备数就越多,完成发现过程所需的帧数就越少。

无反馈帧时隙(FSA_noFBP)算法和有反馈帧时隙(FSA_FBP)算法的对比如图 2 所示,使用无反馈帧时隙算法比使用有反馈帧时隙算法时的 \overline{AF} 值大很多,特别是在邻居设备比较多时尤为明显。这是因为用无反馈帧时隙算法要竞争的设备数不变,从而导致更高的帧内冲突,完成发现过程所需的帧数更多。因此,请求设备在每帧之后发送一个 FBP 的开销可以获得更小的平均帧数。

帧时隙(FSA)算法和争用树算法(CTA)的对比也如图 2 所示,当帧长度较小时,争用树算法的 \overline{AF} 值远小于帧时隙法的 \overline{AF} 值,且帧时隙算法的 \overline{AF} 值趋于无穷大。然而,当帧长度增加到一个依赖于竞争邻居设备数的阈值时,争用树算法的 \overline{AF} 值大于有反馈帧时隙算法的 \overline{AF} 值。对于每种算法,时隙内的冲突随着帧时隙数的增多而减少,因此,完成发现过程所需的平均帧数也减少。然而,争用树算法为每个时隙内的冲突分配一个新帧,使冲突可能被蔓延到更多的时隙。因此,当帧长度很大时,争用树算法比帧时隙算法产生更多的竞争帧数。

(2) 请求设备的平均能耗

整个发现过程中请求设备的平均能耗用 \overline{E}_R 表示,图 3 是对于不同算法且考虑两种不同的 n 值($n=10$ 或 $n=30$)的平均能耗随着帧长度变化的曲线。无反馈帧时隙算法 \overline{E}_R 的表达式对应式(25)中的 $\overline{E}_R(FSA_noFBP)$,有反馈帧时隙算法 \overline{E}_R 的表达式对应式(26)中的 $\overline{E}_R(FSA_FBP)$,争用树算法对应式(27)中的 $\overline{E}_R(CTA)$ 。

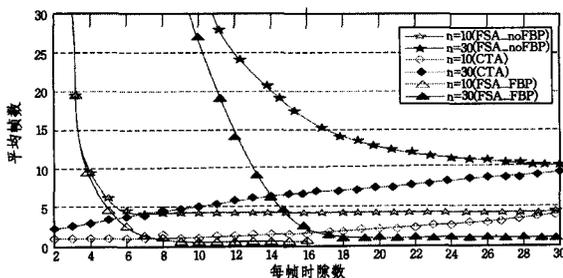


图 3 发现过程请求设备的平均能耗

对于帧时隙算法,当时隙数比较少时,设备产生冲突的概率

率大,所需要的帧数多,从而能耗大;且无反馈帧时隙算法 \overline{E}_R 的值总是大于或等于有反馈帧时隙算法的。随着帧长度的增加, \overline{AF} 的值趋于一个常量。然而,对于更长的一段时间,请求设备需要在每帧的更多时隙中侦听信道,从而导致更多的能耗。因此,当帧长度大于对应的阈值时, \overline{E}_R 的值仍然不断增大,尤其是使用争用树算法时,曲线倾斜率更大,表现出更多的能量增加。

(3) 邻居设备的平均总能耗

在工作循环模式和连续模式下所有邻居设备的平均总能耗用 \overline{E}_A 表示,图 4 和图 5 是对于不同的算法且考虑两种不同 n 值时 \overline{E}_A 随着帧长度变化的曲线。

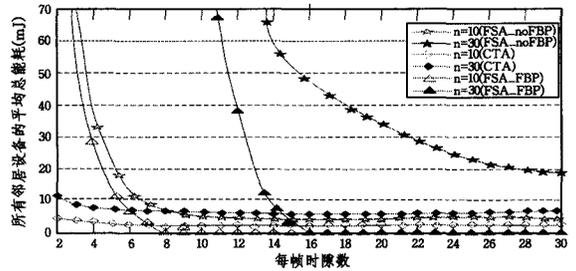


图 4 在工作循环模式下邻居设备的平均总能耗

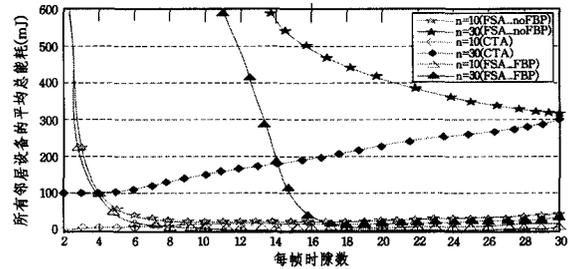


图 5 在连续模式下邻居设备的平均总能耗

从图 4 和图 5 观察到,不管是在工作循环模式下还是在连续模式下,当邻居设备总数 $n=10$ 、帧长度最小值为 2 时,无反馈帧时隙算法和有反馈帧时隙算法的 \overline{E}_A 值趋于无穷大。且所有算法的工作循环模式下的最小 \overline{E}_A 值都小于相应算法的连续模式下的 \overline{E}_A 值,这是因为邻居设备处于睡眠状态比处于接收状态的能耗低。

如上节分析,有反馈帧时隙算法的 \overline{E}_A 值小于无反馈帧时隙算法的 \overline{E}_A 值,因为有反馈帧时隙算法的邻居设备一旦被发现就停止发送 ID 包,从而使邻居设备产生更少的能耗,且平均帧数随着帧内竞争邻居设备数的减少而减小,总能耗随之减少。随着帧长度的增加,两种帧时隙算法的 \overline{E}_A 值趋于一个常数值,且帧长度越大, \overline{E}_A 的值就越小。

如图 4 和图 5 所示,当帧长度较小时,争用树算法的 \overline{E}_A 值是有限的,且远小于帧时隙算法的 \overline{E}_A 值。然而,当帧长度大于一个特定的值时,争用树算法的 \overline{E}_A 值开始大于帧时隙算法的 \overline{E}_A 值,这是因为当帧长度较大时,争用树算法的 \overline{AF} 值高于帧时隙算法的 \overline{AF} 值。在争用树算法的工作循环模式下,当帧长度增加时, \overline{E}_A 的值趋于一个常数;然而在争用树算法的连续模式下, \overline{E}_A 的值随着帧长度线性且成比例地增加,这是因为无竞争时隙中邻居设备能耗增加,且在时隙 $m_{opt}=3$ 时,

(下转第 112 页)

[6] Ker A D. A fusion of maximal likelihood and structural steganalysis[C]//9th International Workshop. Springer Berlin Heidelberg, 2007; 204-219

[7] Kodovsky J, Fridrich J. Steganalysis of JPEG images using rich models[C]// Proceedings SPIE, The International Society for Optical Engineering, 2012; 1-13

[8] Filler T, Pevny T, Bas P. BOSS(Break Our Steganography System)[OL]. <http://www.agents.cz/boos>

[9] Fridrich J, Kodovsky J. Rich models for steganalysis of digital images[J]. IEEE Transactions on Information Forensics and Security, 2011, 7(3): 868-882

[10] Kodovsky J, Fridrich J, Holub V. Ensemble classifiers for steganalysis of digital media[J]. IEEE Transactions on Information Forensics and Security, 2012, 7(2): 432-444

[11] Holub V, Fridrich J. Digital image steganography using universal distortion[C]// 1st ACM Information Hiding and Multimedia Security Workshop. Montpellier, France, 2013

[12] Filler T, Judas J, Fridrich J. Minimizing additive distortion in steganography using syndrome-trellis codes[J]. IEEE Transactions on Information Forensics and Security, 2011, 6(3): 920-935

(上接第 72 页)

E_A 的值最小。

结束语 作为一种近年新兴的无线网络技术, ZigBee 网络在森林火灾检测、环境监测、矿井人员定位等环境中得到广泛的应用, ZigBee 网络的节点电池能量的有效利用能延长其生命周期。当前, ZigBee 网络节能主要集中于跨层考虑优化和改进能耗, 或引入某种节点能耗均衡策略。本文采用分布式邻居发现机制研究 ZigBee 网络能耗。该机制使得 ZigBee 网络中的节点在其可访问的区域内及时发现邻居节点, 通过分析和定义在分布式邻居发现算法下 ZigBee 节点在整个发现过程中所需要的平均帧数, 提出 ZigBee 网络能耗模型, 对 ZigBee 网络节点和邻居节点分别在不同算法下的能耗进行详细分析并得出能耗表达式。

在仿真实验中, 比较了 ZigBee 网络节点和邻居节点在不同算法下的工作循环模式和连续模式下设备的能耗, 实验结果表明对于基于分布式邻居发现机制算法的 ZigBee 网络能耗, 当帧时隙数较小时, 在选择争用树算法时能获得较好的节能效果; 当帧时隙数较大时, 选择无反馈帧时隙算法或有反馈帧时隙算法则可减少节点能耗, 以延长 ZigBee 网络的生命周期。

参 考 文 献

[1] Sun J S. Research on energy consumption mechanism based on ZigBee Network[D]. Beijing: Beijing University of Posts and Telecommunications, 2015(in Chinese)
孙建书. ZigBee 网络能耗优化机制的研究与实现[D]. 北京: 北京邮电大学, 2015

[2] Cao J L, Liu W P, Ren Z, et al. Energy-balanced routing algorithm based on cross-layer design for Inter-PAN communications in Zigbee Networks[J]. Computer Science, 2014, 41(8): 144-147(in Chinese)
曹建玲, 刘文朋, 任智, 等. 基于跨层机制的 ZigBee 网络 PAN 间能耗均衡路由算法[J]. 计算机科学, 2014, 41(8): 144-147

[3] Jin R C, Han Q P, Chen Z Q, et al. Calculating method of node energy consumption for ZigBee wireless networks; China, 201110149806[P]. 2011. 09. 14(in Chinese)
金仁成, 韩庆普, 陈真清, 等. ZigBee 无线网络节点的能耗计算方法: 中国, 201110149806[P]. 2011. 09. 14

[4] Schoute F C. Control of ALOHA signalling in a mobile radio

trunking system[C]// International Conference on Radio Spectrum Conservation Techniques. 1980(188); 38-42

[5] Wieselthier J E, Ephremides A, Michaels L A. An exact analysis and performance evaluation of framed ALOHA with capture [J]. IEEE Transactions on Communications, 1989, 37(2): 125-137

[6] Klaus F. RFID handbook: Fundamentals and applications in contactless smart cards and identification [M]. Hardcover, 2003; 61-159

[7] Burdet L A. RFID multiple access methods[OL]. http://www.vs.inf.ethz.ch/edu/SS2004/DS/reports/06_rfid-mac_report.pdf

[8] Cha J R, Kim J H. Dynamic framed slotted ALOHA algorithms using fast tag estimation method for RFID system[C]// Proc. of IEEE CCNC. 2006; 768-772

[9] Vogt H. Efficient object identification with passive RFID tags [M]// Pervasive Computing. Springer Berlin Heidelberg, 2002; 98-113

[10] Vogt H. Multiple object identification with passive RFID tags [C]// 2002 IEEE International Conference on Systems, Man and Cybernetics. IEEE, 2002, 3: 6

[11] Egea-López E, Vales-Alonso J, Martínez-Sala A S, et al. Performance evaluation of non-persistent CSMA as anti-collision protocol for active RFID tags [M]// Wired/Wireless Internet Communications. Springer Berlin Heidelberg, 2007; 279-289

[12] Capetanakis J. Tree algorithms for packet broadcast channels [J]. IEEE Transactions on Information Theory, 1979, 25(5): 505-515

[13] Janssen A, Jong M. Analysis of connection tree algorithms[J]. IEEE Transactions on Information Theory, 2000, 46(6): 2163-2172

[14] Su X, Xiao Y. Analysis of energy consumption for multiple object identification system with active RFID tags[J]. Wireless Communications and Mobile Computing, 2008, 8(7): 953-962

[15] Law C, Lee K, Siu K Y. Efficient memoryless protocol for tag identification[C]// Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications. ACM, 2000; 75-84

[16] Namboodiri V, Gao L. Energy-aware tag anti-collision protocols for RFID systems [J]. IEEE Transactions on Mobile Computing, 2010, 9(1): 44-59