

低功耗有损网络中的能量均衡高效路由协议

姚玉坤, 杨及开

(重庆邮电大学 移动通信技术重庆市重点实验室, 重庆 400065)

摘 要: 在根据路径瓶颈节点寿命选择最优父节点的低功耗有损网络路由协议(RPL)中,待加入节点未考虑其他节点产生的流量对父节点产生的影响,且在节点加入网络后会导致路径瓶颈节点发生变化。针对上述问题,提出一种基于能量均衡的 RPL 高效路由算法(EBHE-RPL)。在节点加入网络之前,采用目的地通告确认消息分类发送机制减少父节点发送的控制包数量,并通过父节点流量累计机制使计算出的瓶颈节点寿命值与实际值更接近,从而使待加入节点能更准确地确定最优父节点。在节点加入网络后,采用路径瓶颈节点再预估机制,避免后续节点误选瓶颈节点的情况。仿真结果表明,与 RPL 和 EB-RPL 算法相比,EBHE-RPL 在延长网络生存时间、均衡网络能量方面性能较好。

关键词: 瓶颈节点;最优父节点;低功耗有损网络路由协议;能量均衡;网络生存时间

中文引用格式:姚玉坤,杨及开.低功耗有损网络中的能量均衡高效路由协议[J].计算机工程,2016,42(11):8-14.

英文引用格式:Yao Yukun, Yang Jikai. Energy-balanced and High-efficiency Routing Protocol for Low-power and Lossy Networks[J]. Computer Engineering, 2016, 42(11):8-14.

Energy-balanced and High-efficiency Routing Protocol for Low-power and Lossy Networks

YAO Yukun, YANG Jikai

(Chongqing Key Lab of Mobile Communication Technology,

Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

【Abstract】 In existing Routing Protocol for Low-power and Lossy Networks (RPL) which selects the best parent according to expected lifetime of bottleneck node on the path, the joining nodes do not consider the traffic generated by the other sub-nodes of the same parent node or the change of bottleneck node after the node's joining. Aiming at these problems, this paper presents an Energy-balanced and High-efficiency RPL (EBHE-RPL). Firstly, it uses a classifying sending mechanism for DAO-ACK messages to reduce the number of control packets sent by the parent node. Secondly, it uses a traffic accumulation mechanism to calculate the Expected Life Time (ELT) of bottleneck node on each path, so that the calculated life time is closer to the actual value. Finally, it uses a bottleneck re-estimating mechanism after node's joining to avoid the bottleneck's change. EBHE-RPL can improve the accuracy for selecting the best parent and balance the energy of networks. Simulation results show that, compared with RPL and EB-RPL algorithms, the performance of EBHE-RPL is improved in the extension of network lifetime and the balance of network energy.

【Key words】 bottleneck node; best parent node; Routing Protocol for Low-power and Lossy Networks (RPL); energy balance; network life time

DOI:10.3969/j.issn.1000-3428.2016.11.002

0 概述

物联网技术^[1]是现有的通信网和互联网的拓展和延伸,其利用大量的传感设备和其他设备对真实的世界进行感知识别,然后上传到网络,通过网络之

间的相互连接实现物物相连。然而传统的路由协议很难较好地工作在低功耗有损网络中,为了使物联网中每个微型设备都能通过统一的网络层协议与 Internet 连接,国际互联网任务工作组的低功耗路由算法(Routing over Lossy and Low-power Networks,

基金项目:重庆市自然科学基金(CSTC2012JJA40040)。

作者简介:姚玉坤(1964—),女,教授,主研方向为移动通信网络、无线自组织网络;杨及开(通讯作者),硕士研究生。

收稿日期:2015-11-26 **修回日期:**2016-01-13 **E-mail:**yangjikaiabc@163.com

ROLL)工作组从 2010 年开始陆续完成了低功耗有损网络路由协议(Routing Protocol for Low-power and Lossy Networks, RPL)标准,如前期各种草案^[2-3]及 RFC6550^[4]。该协议目前已被广泛应用到多种物联网工程项目中,并引起了学术界的广泛关注。文献[5]针对 P2P-RPL 算法进行了平台的仿真实验,验证了 RPL 路由算法的高效性;文献[6]针对 RPL 协议进行了详细介绍,并通过硬件平台实现了 RPL 组网;文献[7-8]介绍了 RPL 协议的原理,说明该协议已成为物联网技术的基础知识。

由于低功耗有损网络^[9]是由功率、存储空间、处理能力等资源受限的嵌入式设备所组成的网络,它们可以通过多种链路如 IEEE 802.15.4、蓝牙、低功耗 WiFi 等连接。这些嵌入式设备往往能量有限,且它们之间的链路,很容易因为能量耗尽而失效,从而使数据不能及时或者是不能传输到目的节点。因此能量均衡成为低功耗有损网络的重要研究方向。文献[10]在 RPL 标准路由协议的基础上,提出了考虑节点剩余能量的能效 RPL 算法(Energy Efficient RPL, EERPL),在路径选择时综合考虑跳数和节点能量两方面因素,让数据尽可能通过能量充裕的节点传输,以达到均衡网络能量和延长网络生存时间的目的。文献[11-12]提出将预计传输次数和节点剩余能量相结合的父节点选择策略,使节点选择综合条件最好的节点加入面向目的地的有向无环图(Destination Oriented Directed Acyclic Graph, DODAG),从而均衡各节点的能耗。文献[13]提出将链路质量、节点剩余能量和节点的传输功率相结合的 L^2AM 参数作为路由度量来选择最优父节点,可均衡网络能量并延长网络生存时间。文献[14]提出的 EB-RPL 算法,其设计思想是待加入节点计算自身通过各个父节点加入 DODAG 后各路径上瓶颈节点的预计寿命,并进行比较,选择预计寿命值最大的瓶颈节点所在路径上的父节点作为最优父节点。

文献[14]算法能一定程度地均衡各路径瓶颈节点的能量,延长网络生存时间,但该算法未考虑父节点除待加入节点以外的其他子节点产生的流量对计算路径瓶颈节点预计寿命的影响,会使待加入节点计算出的各路径瓶颈节点预期生存时间偏长,影响节点判断路径瓶颈节点的准确性。针对该问题,本文提出基于能量均衡的 RPL 高效路由算法(Energy-balanced and High-efficiency RPL, EBHE-RPL)。

1 RPL 路由协议

RPL 协议是一种复杂的路由协议,主要由 DODAG 构建、路由过程、节点工作模式、修复机制、

安全机制、多播运行策略和可管理性等部分组成^[4]。本节简要介绍其中的 DODAG 构建和路由过程。

1.1 DODAG 构建

RPL 是一种距离矢量路由协议,节点可以沿着一个以边界节点或网关节点为根节点的 DODAG 来进行路由选择。一种 DODAG 的构建过程如图 1 所示。

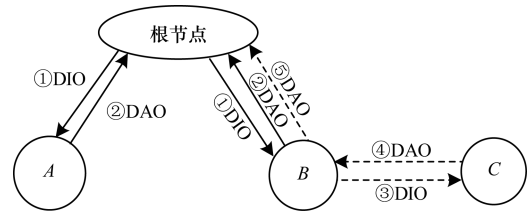


图 1 DODAG 构建过程

在图 1 中,边界节点或根节点通过周期性地广播 DODAG 信息对象(DODAG Information Object, DIO)消息(如①DIO)来构建和维护 DODAG, DIO 消息用于待加入节点发现 RPL Instance、了解节点自己的配置参数、选择 DODAG 父代集以及维护 DODAG 的信息。待加入节点 A、B 收到①DIO 消息后,将根节点作为自己的父节点,然后更新 DIO 消息中的相关信息,再根据 Trickle 机制^[15]广播 DIO 给其邻居节点。当 B 的邻居节点 C 收到 B 广播的③DIO 消息后,先计算自身的 Rank 值,然后与节点 B 的 Rank 值进行比较,如果节点 C 的 Rank 值大于节点 B 的 Rank 值,则将节点 B 加入到父节点集,最后从父节点集中根据目标函数选择一个最佳的节点作为自己的最优父节点并加入到 DODAG。所有节点重复上述加入和广播操作,直到网络中所有节点都加入 DODAG。当节点加入 DODAG 后就建立起了一条从自身到根节点的上行路径。

待加入节点也可以根据根节点在 DIO 消息中的配置信息来决定是否向父节点发送目的地通告(Destination Advertisement Object, DAO)消息来建立一条从根节点到自身的下行路径。如图 1 中, A、B、C 节点均向父节点发送的 DAO 消息。

在存储模式中,父节点 B 收到子节点 C 的 DAO 消息后,会提取④DAO 消息中的目的地址信息,并将该地址信息存储在路由表中,然后将自己的地址信息添加到 DAO 消息的地址信息中作为地址前缀,最后将新的⑤DAO 消息发送给父节点,这样根节点和每个路由节点就可以得到通往子孙节点的下一跳信息。

在非存储模式中,节点不存储任何节点的路由信息,收到 DAO 消息的节点直接转发该 DAO 消息给自己的父节点,直到这些 DAO 消息到达根节点,根节点收到各个节点的 DAO 消息后就可以计算出

从根节点到各个节点的路径信息。

1.2 数据传输的路由过程

RPL 路由协议的数据传输有 3 种路由方式,分别为 MP2P 路由、P2MP 路由和 P2P 路由。

如前文所述,每个节点加入 DODAG 之后就有一条从自身到根节点的上行路径。在数据传输阶段,各节点默认将数据转发给最优父节点,如此反复直到数据传输到根节点。这种由多个节点向一个根节点传输的数据传输的方式称为 MP2P 路由。

根节点可以根据各个 DAO 消息来确认每个节点的路由信息,然后根据这些数据传输信息将数据转发给多个节点,这种数据传输方式称为 P2MP 路由。

节点也可以将数据传输到除根节点以外的某一特定的目的节点,这种数据传输方式称为 P2P 路由。P2P 路由因节点工作模式的不同而不同。

如图 2 所示,在存储模式中,节点将数据包往上传输,直到到达目的节点与节点自身的共同祖先节点,再由共同祖先节点查找到发往目的节点的下一跳进行数据转发。在非存储模式中,源节点需要先将数据传输至根节点,再由根节点将数据传输至目的节点。

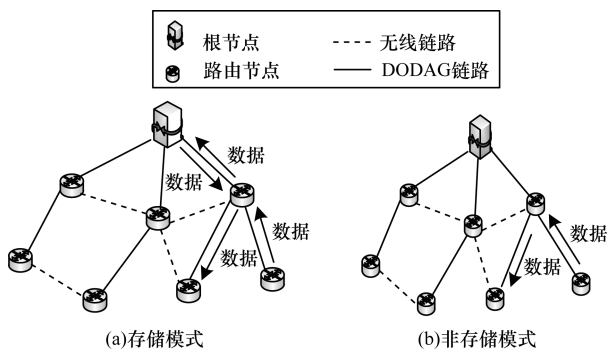


图 2 RPL 协议 P2P 路由示意图

2 网络模型及问题描述

2.1 网络模型及定义

设有一个具有 N 个随机部署的无线传感器节点形成的低功耗有损网络,并给出如下假设:

1) 在 EBHE-RPL 算法中,除边界节点外的所有节点随机部署在一个 $300\text{ m} \times 300\text{ m}$ 的正方形监测区域,边界节点位于矩形边缘。所有的节点一旦布置完成,位置不发生移动。

2) 每个节点休眠时间较短,都能监听到各邻居节点的发包情况。

3) 网络中包含边界节点或根节点、路由节点和叶子节点,其中根节点能量无限,路由节点和叶子节点的能量由电池提供,能量消耗完成后不再给予补充。

4) 路由节点既可以产生数据分组又可以对数据分组进行转发,而叶子节点只能产生数据分组,不能转发其他节点的数据分组。所有节点的传输范围均为 R 。

定义 1 瓶颈节点:表示节点到根节点的传输路径上剩余能量最小的节点。

定义 2 预期生存时间 (Expected Life Time, ELT):表示通过估算方式得到的节点预计生存时间值,其计算公式^[14]为:

$$ELT(n) = \frac{E_{\text{res}}(n) \times \text{DATA_RATE}}{T_{\text{total}}(n) \times ETX(n, P_n) \times P_{\text{tx}}(n)} \quad (1)$$

其中, $E_{\text{res}}(n)$ 代表节点剩余能量; DATA_RATE 代表数据包发送速率; $ETX(n, P_n)$ 代表节点到其父节点的链路预期传输次数; $P_{\text{tx}}(n)$ 代表节点的发射功率; $T_{\text{total}}(n)$ 代表节点转发给父节点的数据流量,其计算公式为:

$$T_{\text{total}}(n) = T_{\text{gen}}(n) + \sum_{i \in \text{Children}(n)} T_{\text{total}}(i) \quad (2)$$

其中, $T_{\text{gen}}(n)$ 代表节点产生的数据流量; $T_{\text{total}}(i)$ 为同一路径上第 i 个子节点的数据流量和。

2.2 问题描述

文献[14]提出的 EB-RPL 算法通过计算得出每条路径瓶颈节点的 ELT 值,再选出 ELT 值最大的那个瓶颈节点所在路径上的父节点作为最优父节点,以此来完成节点加入到 DODAG 和拓扑构建的步骤。但笔者通过深入研究,发现该算法存在以下 3 个问题:

1) 待加入节点在计算路径瓶颈节点的 ELT 值时,只考虑了同一路径上的节点对瓶颈节点的影响,没有考虑父节点除待加入节点以外的其他子节点产生的流量对瓶颈节点的影响,使计算得出的路径瓶颈节点的预期生存时间偏长,导致待加入节点判断路径瓶颈节点的准确性受到影响,不利于均衡各路径瓶颈节点的能量。

2) 在同一路径上只考虑路径当前瓶颈节点和待加入节点自身的 ELT 值大小,在这两个节点中选一个节点作为瓶颈节点并通知邻居节点,没有考虑节点加入 DODAG 后路径瓶颈节点的变化情况,使后续待加入节点不能准确地选择最优父节点,不能有效地达到均衡能量和延长网络生存时间的效果。

3) 未考虑不同路径瓶颈节点 ELT 值相同的情况,如果盲目选择,可能会加重某些节点的负担,不利于均衡。

3 EBHE-RPL 算法

本文针对 EB-RPL 算法存在的问题,提出了改进算法 EBHE-RPL,该算法的主要贡献是设计了 3 个新机制,分别是目的地通告确认消息分类发送机

制、父节点流量累计机制和路径瓶颈节点再预估机制。

3.1 EBHE-RPL 算法的新机制

3.1.1 目的地通告确认消息分类发送机制

目的地通告确认消息分类发送机制是指父节点根据子节点发送的 DAO 消息所代表的不同具体含义来决定是否向子节点发送 DAO-ACK 的机制。通过该机制,可以禁止父节点因为收到子节点发送的 no-path DAO 消息而发送 DAO-ACK 消息,能减小控制开销。该机制的具体步骤如下:

步骤 1 子节点向父节点单播 DAO 消息通知其在当前 DODAG 中的状态,如果该 DAO 消息是为了通告父节点其目的,则把 DAO 消息的 K 标志位置 1,表示节点需要父节点反馈一个 DAO-ACK 消息;如果该 DAO 消息是为了告知父节点其 no-path 的信息,则将 DAO 消息的 K 标志位置 0,表示节点不要求父节点反馈 DAO-ACK 消息。

步骤 2 父节点收到子节点发送的 DAO 消息后,根据该 DAO 消息的 K 标志位来判断节点是否发送 DAO-ACK 消息。如果 $K = 1$,则向子节点单播一个 DAO-ACK 消息,反之,节点不发送 DAO-ACK 消息。

3.1.2 父节点流量累计机制

父节点流量累计机制是指在待加入节点加入 DODAG 前,将父节点的其他子节点通过它加入 DODAG 后产生的数据流量累加起来计算瓶颈节点预期寿命的机制。以图 3 为例,在节点 G 选择 F 节点作为最优父节点的过程中, F 节点还有另外一个先于 G 节点加入 DODAG 的 H 节点, H 节点的数据流量也会减少瓶颈节点 C 的 ELT 值,因此,在 G 计算瓶颈节点 ELT 值时,必须将 H 节点的数据流量计算进来。

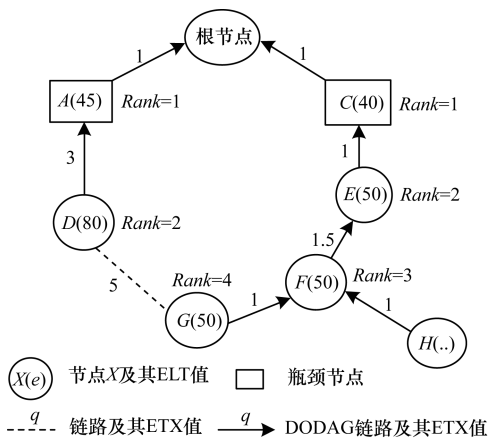


图 3 父节点流量累计机制示意图

父节点流量累计机制的具体实现步骤如下:

步骤 1 节点通过跨层监听方式监听邻居节点的发包情况,如果监听到 DAO-ACK 消息,则在邻居表中记录相应节点发送的 DAO-ACK 的个数,假设记为 $C_{\text{DAO-ACK}}$ 。

步骤 2 当待加入节点收到 DIO 消息时,首先计算 Rank 值,然后将比自身 Rank 小的节点加入父节点集。

步骤 3 计算各路径瓶颈节点 ELT 值。节点用监听到父节点发送的 DAO-ACK 消息数表征为该父节点的其他子节点的个数,并将这些子节点的流量和待加入节点通过该父节点加入后将要发送的数据流量累加起来计算各父节点所在路径瓶颈节点 ELT 值。通过瓶颈节点转发的数据流量计算公式为:

$$T_{\text{total}}(B) = T_{\text{gen}}(B) + \sum_{i \in \text{Children}(B)} T_{\text{total}}(i) \quad (3)$$

其中, $T_{\text{total}}(B)$ 表示通过瓶颈节点的数据流量; $T_{\text{gen}}(B)$ 表示瓶颈节点自身产生的数据流量; $T_{\text{total}}(i)$ 为同一路径上第 i 个子节点发送的数据流量; B 表示路径上的瓶颈节点即预计正常工作时间最短的节点。节点在单位时间内传输 1 bit 数据的能量消耗为:

$$ETX(B, P_B) \times P_{\text{tx}}(B) \quad (4)$$

其中, ETX 表示节点将数据发送给其父节点的平均发送次数; P_B 为节点 B 的父节点; P_{tx} 为传输功率。假设瓶颈节点 B 当前的剩余能量为 $E_{\text{res}}(B)$,则节点通过某父节点加入到 DODAG 之后瓶颈节点的预期生存时间为:

$$ELT(B) = \frac{E_{\text{res}}(B)}{T'_{\text{total}}(B) \times P_{\text{tx}}(B)} \times \text{DATA_RATE} \quad (5)$$

其中:

$$T'_{\text{total}}(B) = (C_{\text{DAO-ACK}} \times T_{\text{gen}}(n) + T_{\text{total}}(B)) \times ETX(B, P_B) \quad (6)$$

父节点流量累计机制通过监听方式间接得到通过当前父节点的流量,使待加入节点计算到的瓶颈节点 ELT 值更加准确,有利于待加入节点准确地判断路径瓶颈节点和选择最优父节点。此外,该机制没有增加任何网络开销。

3.1.3 路径瓶颈节点再预估机制

路径瓶颈节点再预估机制是指在节点加入 DODAG 之后,预估自己在后续节点通过它加入 DODAG 后是否会因为转发后续节点的数据包而比原来的瓶颈节点更有可能成为路径新的瓶颈节点的机制,如图 4 所示。

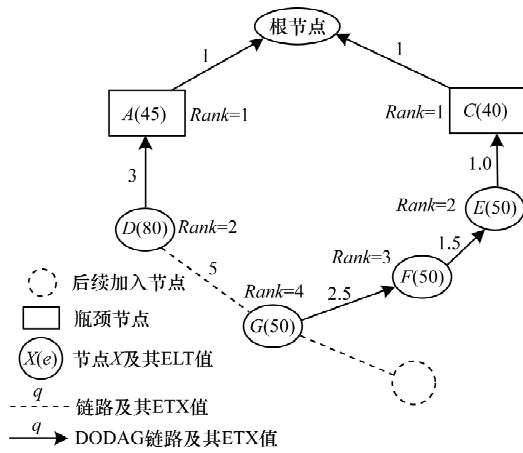


图4 路径瓶颈节点再预估机制示意图

假设节点 G 当前通过 F 节点加入 DODAG, 此时 ELT_C 为 35 单元, ELT_G 为 37.5 单元, 因此, C 还是路径瓶颈节点, G 的后续节点也会将 C 作为瓶颈节点。但节点 G 与 F 之间的链路质量较差, 在节点 G 的后续节点通过 G 加入 DODAG 之后, ELT_C 变为 30 单元, 而 ELT_G 却减小为 25 单元, G 才应该成为新的瓶颈节点, 因此, 后续节点应该以节点 G 作为路径瓶颈节点并和自身的 ELT 作比较。

实现路径瓶颈节点再预估机制的操作步骤如下:

步骤 1 节点在加入 DODAG 后, 先判断自身是否是叶子节点, 如果是叶子节点, 则放弃再预估过程, 否则, 转到步骤 2。

步骤 2 若节点加入 DODAG 时, 自身是 ELT 值最大的瓶颈节点, 则直接广播 DIO 消息给邻居节点, 放弃再预估过程。否则, 转到步骤 3。

步骤 3 假设后续还有一个待加入节点加入 DODAG, 估算节点再转发一个数据包到根节点后自身和瓶颈节点的 ELT 值大小, 分别记为 $ELT(n)^{new}$ 和 $ELT(B)^{new}$ 。若 $ELT(n)^{new} < ELT(B)^{new}$, 则节点自身成为当前路径最新的瓶颈节点, 并将自身 ELT 值和 ETX 参数封装到 DIO 消息中, 广播给其邻居节点; 若 $ELT(n)^{new} > ELT(B)^{new}$, 则自身放弃成为瓶颈节点, 节点发送给其他节点的 DIO 消息中的参数仍为当前路径瓶颈节点的参数。

瓶颈节点再预估机制可以尽量避免后续节点误选瓶颈节点的情况。

3.2 EBHE-RPL 算法的操作步骤

下面给出 EBHE-RPL 算法的执行操作步骤:

步骤 1 网络初始化时, 按照 DAO-ACK 消息分类发送机制进行全网节点的配置工作。

步骤 2 根节点周期性广播带有配置信息的 DIO 消息。

步骤 3 收到 DIO 消息的节点判断是否是第一

次收到该 DIO 消息, 若不是, 则丢弃该 DIO 消息, 否则, 按照父节点流量累计机制计算各父节点所在路径上瓶颈节点的 ELT 值, 并执行步骤 4。

步骤 4 确认各路径瓶颈节点。待加入节点计算到各路径瓶颈节点的 ELT 值后, 再计算自己的 ELT 值, 并将该值与加入该路径后瓶颈节点的 ELT 值进行比较, 节点自身 ELT 值的计算如下:

$$ELT(n) = \frac{E_{res}(n) \times DATA_RATE}{T_{total}(n) \times ETX(n, P_n) \times P_{tx}(n)} \quad (7)$$

如果 $ELT(n) < ELT(B)$, 则将节点本身作为当前路径上的最新瓶颈节点; 反之, 当前路径瓶颈节点不做变化。

步骤 5 选择最优父节点。将计算到的各路径瓶颈节点的 ELT 值进行比较, 选择瓶颈 ELT 值最大的路径上的父节点作为最优父节点。若出现两条或多条路径上 $ELT(B)$ 值相同的情况, 应当考虑局部因素, 待加入节点优先选取发送 DAO-ACK 消息最少的父节点作为最优父节点, 以减小父节点的计算开销和存储开销, 均衡各父节点因发送控制包所产生的能耗。

步骤 6 节点加入 DODAG 后, 按照路径瓶颈节点再预估机制对 DIO 消息中的路径瓶颈节点的信息进行更新操作, 再将新的 DIO 消息广播给邻居节点。

4 仿真实验与结果分析

本文采用 contiki^[16] 操作系统的 COOJA^[17] 软件进行仿真平台搭建, 同时选取 RPL 协议和 EB-RPL 算法作为比较对象, 通过仿真比较分析其与 EBHE-RPL 算法在网络生存时间、总发包数量、控制报数量和节点能耗等性能指标上的差异。

4.1 网络场景及参数设置

为了和文献[14]的 EB-RPL 算法对比, 同样将网络中所有节点的工作模式设为存储模式。无线信道模型采用阴影衰落模型。本文实验在网络场景中分别放置 20 个 ~ 90 个节点, 每种场景分别仿真 20 轮, 具体仿真参数如表 1 所示。

表 1 仿真参数设置

参数名	参数值
仿真周期/s	3 600, 7 200
节点数	20 ~ 90
场景面积/m ²	300 × 300
负载类型发包速率/(pkt · min ⁻¹)	CBR1
数据包尺寸/Byte	127
MinHopRankIncrease	256
MAC 协议	IEEE802.15.4
节点模式	存储模式
信道阴影衰落因子	1.97
标准偏差	2.0

本文将以 ETX 为路由度量的 RPL 协议、EB-RPL 算法和本文提出的 EBHE-RPL 算法的仿真参数设置为如表 1 所示参数值,上述 3 种算法均支持局部修复和全局修复策略。

4.2 仿真结果及分析

4.2.1 网络生存时间

网络生存时间即网络从初始运行到出现第一个死亡节点的这段时间,反映了网络中所有节点能有效运行的时间,是衡量网络拓扑性能的一项重要指标。3 种算法的网络生存时间比较如图 5 所示。



图 5 网络生存时间比较

由图 5 可知,2 种改进算法都要比以 ETX 为参数度量的 RPL 协议的网络生存时间要长。因为单纯以 ETX 作为参数选择最优父节点只考虑了节点与父节点之间的链路质量,虽然总的传输次数会比较少,但是没有考虑路径上节点的最小剩余能量,不能起到能量均衡和延长网络生存时间的作用。

当网络中节点数目小于 50 时,本文提出的 EBHE-RPL 算法则要比 EB-RPL 算法性能要好 10% 左右,这是因为在计算瓶颈节点的 ELT 时,待加入节点将父节点的其他子节点的负载考虑进来,得到的 ELT 值会小于或等于通过 EB-RPL 算法得到的 ELT 值,待加入节点能够更明确地知道选择哪个节点才能达到负载均衡、延长网络生存时间的最佳效果。

4.2.2 总事件数

总事件数指的是在网络生存期内,根节点统计到的下游节点向上发送的数据包数量。总事件数与网络生存时间成正比关系,生存时间越长,总事件数越大。3 种算法在事件数方面的仿真结果如图 6 所示。由图 6 可知,随着节点数量的增加,3 种算法的事件数在增加,当节点数为 60 个节点以后,因为网络中节点数量过多而导致网络生存时间急剧下降,因此,网络的事件数也在急剧减少。EBHE-RPL 算法的事件数比其他 2 种算法的事件数多的原因是该算法在其他 2 种算法的基础上延长了网络的生存时间,因此,在生存期内网络中节点能向根节点发送数据包的数量会比另外 2 种算法的数量多。

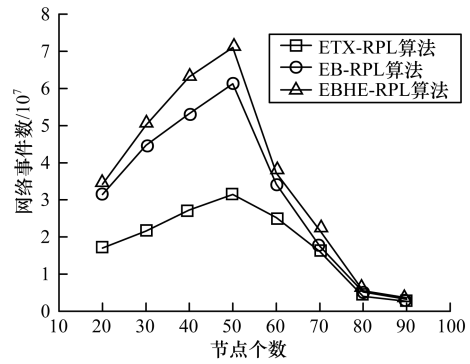


图 6 网络事件数比较

4.2.3 通告消息确认控制包的数量

通告消息确认包的数量指的是在网络生存期内所有节点向根节点发送 DAO 消息后收到的 DAO-ACK 消息包的数量统计平均值。该值越大,说明算法的控制开销越大。3 种算法在此方面的仿真结果如图 7 所示。

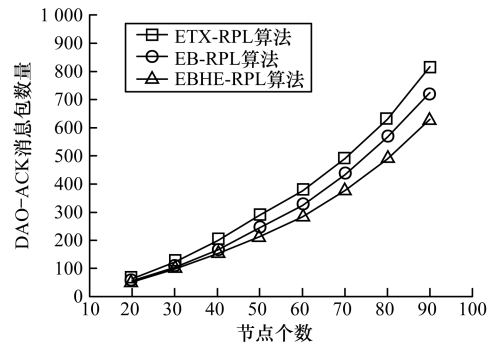


图 7 DAO-ACK 消息包数量比较

由图 7 可知,随着节点个数增加,DAO-ACK 消息也在增加,EBHE-RPL 算法产生的 DAO-ACK 消息包数量最少,主要是因为该算法阻止了少数节点因为 No-Path DAO 消息获得的 DAO-ACK 消息,网络中所有节点发送的 DAO-ACK 消息全为目的地通告确认包,而非路径失效通告确认包,因此,网络中的 DAO-ACK 消息包的数量会减少。

4.2.4 节点能耗

本文测试节点能耗是通过检测在 50 个节点的网络模型中,距边界路由器不同距离的节点在仿真 1 h 和仿真 2 h 的平均能量消耗来获得的。节点能耗结果如图 8 所示。由图 8 可见,EBHE-RPL 算法比 EB-RPL 算法和 RPL 协议在节点能耗方面会低一点,因为 EBHE-RPL 算法减少了不必要的 DAO-ACK 消息的发送过程,节省了因这些 DAO-ACK 消息包所带来的控制开销。另外,在仿真到 2 h 后,节点的能耗明显要比仿真 1 h 的能耗 2 倍大,这是因为节点除了有发送能耗外,还有 CPU 能耗和监听能耗,所以会出现比仿真 1 h 能耗的两倍大很多的情况。

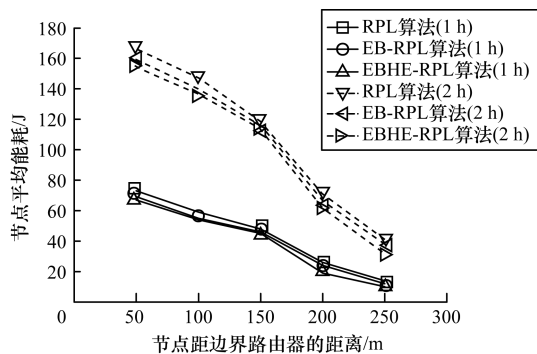


图8 仿真1 h和2 h后的节点平均能耗比较

5 结束语

本文提出一种基于能量均衡的RPL路由算法EBHE-RPL,结合RPL能量均衡相关算法的特性,改进现有算法选择最优父节点的计算方法,提高算法能量均衡和延长网络生存时间的能力。同时针对中间节点因链路质量问题而出现比瓶颈节点先死亡的情况加以改进,使后续子孙节点能够根据当前各个父节点相关的路径情况,选择最佳的父节点,起到延长网络生存时间的作用。理论分析和仿真结果表明,相对ETX-PRL和EB-PRL算法,EBHE-RPL算法增加了最优父节点选择的父节点流量累计机制和瓶颈节点再预估机制,在网络能量均衡和延长网络生存时间方面,本文算法性能较优。

参考文献

[1] 刘云浩. 物联网导论[M]. 北京: 科学出版社, 2010: 3-11.

[2] IETF W G. Routing over Low Power and Lossy Networks [EB/OL]. [2015-10-20]. <https://datatracker.ietf.org/wg/roll/documents/>.

[3] Winter T, Thubert P, Brandt A, et al. RPL: Ipv6 Routing Protocol for Low Power and Lossy Networks [EB/OL]. (2011-03-11). <http://tools.ietf.org/html/draft-ietf-roll-rpl-19>.

[4] IETF ROLL Working Group. RPL: IPv6 Routing Protocol for Low Power and Lossy Networks; IETF Std RFC 6550—2012[S]. IETF, 2012.

[5] Baccelli E, Philipp M, Goyal M. The P2P-PRL Routing Protocol for Ipv6 Sensor Networks; Testbed Experiments [C]// Proceedings of the 19th IEEE Symposium on Computers and Communications. Washington D. C., USA; IEEE Press, 2011: 937-942.

[6] 李凤国. 基于6LoWPAN的无线传感器网络研究与实现[D]. 南京: 南京邮电大学, 2013.

[7] 彭力. 物联网技术应用基础[M]. 北京: 冶金工业出版社, 2011: 171-188.

[8] Gaddour O, Koubaa A. RPL in a Nutshell; A Survey [J]. Computer Networks, 2012, 56(14): 3163-3178.

[9] 李树军. 基于6LoWPAN的RPL路由协议研究[J]. 重庆工商大学学报(自然科学版), 2013, 30(8): 72-77.

[10] 胡芹艳, 尹长川. 无线传感器网络中的RPL路由协议研究[J]. 物联网技术, 2014, 14(1): 57-62.

[11] Chang Linhuang, Lee T H, Chen Shujan, et al. Energy-efficient Oriented Routing Algorithm in Wireless Sensor Networks [C]// Proceedings of IEEE International Conference on Systems, Man, and Cybernetics. Washington D. C., USA; IEEE Press, 2013: 3813-3818.

[12] Abreu C, Ricardo M, Mendes P M. Energy-aware Routing for Biomedical Wireless Sensor Networks [J]. Journal of Network and Computer Applications, 2014, 40(1): 270-278.

[13] Capone S, Accettura N, Accettura N, et al. An Energy Efficient and Reliable Composite Metric for RPL Organized Networks [C]// Proceedings of the 12th IEEE International Conference on Embedded and Ubiquitous Computing. Washington D. C., USA; IEEE Press, 2014: 178-184.

[14] Iova O, Theoleyre F, Noel T, et al. Improving the Network Lifetime with Energy-balancing Routing: Application to RPL [C]// Proceedings of the 7th IFIP Wireless and Mobile Networking Conference. Washington D. C., USA; IEEE Press, 2014: 1-8.

[15] Levis P, Clausen T, Hui J, et al. The Trickle Algorithm [EB/OL]. (2011-07-14). <http://tools.ietf.org/html/draft-ietf-roll-trickle-08>.

[16] Dunkels A, Gronvall B, Voigt T. Contiki—A Lightweight and Flexible Operating System for Tiny Networked Sensors [C]// Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks. Washington D. C., USA; IEEE Press, 2004: 455-462.

[17] Dunkels A, Eriksson J, Finne N, et al. COOJA/MSPSim: Interoperability Testing for Wireless Sensor Networks [C]// Proceedings of the 2nd International ICST Conference on Simulation Tools and Techniques. Rome, Italy; ICST Press, 2010: 716-722.