

能量高效的 ZigBee 网络改进路由策略

张 擎¹, 刘淑美¹, 柴乔林²

(1. 山东大学艺术学院, 济南 250100; 3. 山东大学计算机科学与技术学院, 济南 250016)

摘 要: 在分析现有 ZigBee 网络路由策略的基础上, 以节省网络能耗和均衡网络负载为目的, 提出一种能量高效的改进路由策略。该策略分别对 ZigBee 网络的簇树拓扑结构以及基于簇树算法和 AODVjr 算法的 ZBR 路由策略进行改进。改进后的路由策略在尽量减少路由开销的同时充分考虑对能量偏低节点的保护。仿真实验验证, 改进策略能有效减少网络能耗, 均衡网络负载, 最大化网络的生存时间。

关键词: ZigBee 网络; 路由策略; 能量高效; 负载均衡; OMNET++ 仿真

Energy Efficient Improved Routing Strategy for ZigBee Network

ZHAGN Qing¹, LIU Shu-mei¹, CHAI Qiao-lin²

(1. Fine Arts School of Shandong University, Jinan 250100; 2. School of Computer Science and Technology, Shandong University, Jinan 250016)

【Abstract】 Based on the research of ZigBee Network's present routing strategy, this paper proposes a routing strategy with the goal to save network's energy cost and to balance data loading among nodes. The new strategy improves ZigBee Network's cluster-tree topology structure and the present ZigBee Routing(ZBR) strategy which is based on cluster-tree algorithm and AODVjr algorithm. The strategy tries its best to reduce routing overhead and to protect low energy nodes at the same time. According to simulation results, it proves that strategy can effectively reduce energy cost and balance loading, successfully prolongs network's lifetime.

【Key words】 ZigBee networks; routing strategy; energy efficient; load balancing; OMNET++ simulation

1 概述

为满足无线传感器和小型控制网络中低成本设备的无线联网要求, 基于 IEEE802.15.4 标准的 ZigBee 技术应运而生, 并在家庭保健、医学检测、消费系统及环境检测等领域得到了广泛的应用^[1]。ZigBee 网络路由策略的设计直接影响网络运行效率和服务质量, 已成为目前 ZigBee 网络的研究热点。ZigBee 网络有星型、树形和 Mesh 这 3 种拓扑组织形式。由于能够结合三者优点, 簇树结构网络拓扑被广泛应用。ZigBee 网络通常采用簇树路由算法^[2]和 AODVjr^[3]算法相结合的路由选择策略。簇树算法是静态路由算法, 节点依据簇树拓扑结构进行路由选择, 不需要存储路由表, 但数据并非沿最优路径传输, 造成能量浪费和服务延迟。AODVjr 算法在传统 AODV 算法的基础上发展而来, 通过洪泛寻路消息寻找最优路径, 需要大量的路由开销。ZigBee 网络通常采用将簇树算法和 AODVjr 算法结合的 ZBR 路由策略, 以求使网络满足不同的应用需求, 优化网络性能。但由于两算法固有的缺点, 使 ZBR 策略在节省网络能耗和负载均衡等方面存在不足。

本文以节省网络整体能耗和均衡网络负载为主要研究目标, 分别对 ZigBee 网络簇树拓扑结构和 ZBR 路由策略进行改进。对簇树拓扑结构的改进使簇树结构在节省路由开销和负载均衡方面更加合理。在对 ZBR 策略的改进中, 利用簇树结构减少了额外的路由消耗, 并通过改进对 RREQ 和 RREP 分组的处理方式, 进一步节省路由开销, 保护能量偏低节点, 均衡网络负载。

2 ZigBee 网络层技术简介

2.1 簇树拓扑结构及地址分配

IEEE802.15.4 根据节点的计算能力、存储容量定义了全功能设备(FFD)和精简功能设备(RFD)。各种设备的特点及其

在网络中的功能和角色描述参见文献[4]。

ZigBee 网络一般采用簇树状分层拓扑结构组织节点^[2]。中心协调器首先创建网络, 当一个节点申请加入该网络时, 它选择一个有路由功能的父节点加入, 它们之间就形成了父子关系。每个进入网络的节点都会得到父节点为其分配的一个在该网络中唯一的 16 bit 网络地址, 节点间利用此网络地址进行寻路和相互通信。

规定每个父节点最多可以连接 C_m 个子节点, 这些子节点中最多可以有 R_m 个路由器节点, 网络的最大深度为 L_m , 对同一个网络来说, 不同的节点通常有相同的 R_m , C_m 和 L_m 。 $Cskip(d)$ 是网络深度为 d 的父节点为其子节点分配的地址之间的偏移量。设置协调器的网络地址为 $0x0000$ 以及网络深度 $Depth_0=0$ 。

假设父节点 k 的深度为 d , 地址为 A_k , 则新加入的节点的深度为 $d+1$ 。如果新加入的节点是其父节点的第 n 个 RFD 型终端子节点, 则父节点为该节点分配网络地址:

$$A_n = A_k + Cskip(d) \times R_m + n \quad (1 \leq n \leq C_m - R_m)$$

如果新加入的节点是其父节点的第 n 个 FFD 型路由器子节点, 则父节点为该节点分配网络地址:

$$A_n = A_p + 1 + Cskip(d) \times (n-1) \quad (1 \leq n \leq R_m)$$

其中,

$$Cskip(d) = \begin{cases} 1 + C_m \times (L_m - d - 1) & \text{如果 } R_m = 1 \\ \frac{1 + C_m - R_m - C_m \times R_m^{L_m - d - 1}}{1 - R_m} & \text{其他} \end{cases}$$

作者简介: 张 擎(1982-), 女, 硕士, 主研方向: 网络与分布式技术; 刘淑美, 讲师; 柴乔林, 教授

收稿日期: 2009-11-20 **E-mail:** zq-abby@163.com

2.2 簇树路由算法

簇树路由算法完全依赖簇树结构转发数据。如果终端节点要发送数据包到网络中的其他节点,则直接将该数据包转发给其父节点,由父节点进行转发。

如果一个路由器节点要转发数据包到网络地址为 D 的目的节点,已知该路由器节点的网络地址和深度分别为 A 和 d 。

首先,该路由器节点会依据下述表达式判断目的节点是否是其后裔节点: $A < D < A + Cskip(d-1)$ 。

如果目的节点是其后裔节点,则下一跳节点地址为

$$N = \begin{cases} D & \text{目的节点是其子节点} \\ A+1 + \left\lfloor \frac{D-(A+1)}{Cskip(d)} \right\rfloor \times Cskip(d) & \text{否则} \end{cases}$$

否则,下一跳节点为该节点的父节点。簇树结构路由限制了寻址的灵活性,数据通过较长路径传递,从而浪费网络能耗并且引入服务延迟。

2.3 AODVjr路由算法

AODVjr 算法基于 AODV 算法,通过洪泛 RREQ 分组寻找最优路径。AODVjr 舍弃了 AODV 中的目标序列号和跳数,只能是目标节点来对最先到达的 RREQ 信号做出响应,这种策略也被称作点到点策略。同时 AODVjr 取消了 HELLO 信息的发送,由目标节点定期向 Source 发送 KEEP_ALIVE 连接信息来维持路由。AODVjr 算法虽然能够找到一条最短路径,但洪泛寻址消息和路由维护都需要大量额外开销,最短路径的选择也常常使处在特殊位置的节点快速消耗能量,造成网络能耗的不均衡。

2.4 ZigBee路由策略

ZigBee 路由(ZBR)^[5]策略是为簇树结构 ZigBee 网络设计的路由策略,将节点分为 2 类: RN+和 RN-。RN+是具有足够存储空间和能力执行 AODVjr 协议的节点(FFD 节点),RN-是存储能力和计算能力受限的终端节点(RFD 节点)。RN-收到一个数据分组后只能用簇树算法处理。RN+收到分组后,发起 AODVjr 中的路由发现过程,找到一条通往目的节点的最短路径,如果某条链路发生中断,RN+节点将发起本地修复过程修复路由。

3 改进路由策略

3.1 节点能量分区

在 ZigBee 网络中设 2 个临界值 $\alpha, \beta (\alpha > \beta)$,依据节点的剩余能量,2 个临界值将节点划分到 3 个不同的区域:

(1)充足区:如果节点当前剩余能量值大于 α 则位于充足区。新节点加入时优先考虑该区域节点成为父节点;路由算法中应优先选择包含充足区节点多的路径。

(2)偏低区:如果节点当前剩余能量介于 α 和 β 之间,它就位于偏低区。该区域的路由器可以参与路由,但在路由选择时应尽量少地选择经过此类节点的路径。

(3)警戒区:如果节点当前剩余能量小于 β ,则位于警戒区。新节点加入时尽量避免选择警戒区节点为父节点;如果节点是路由器,则不再参与寻路过程,只转发目的节点是自己后代的数据分组。

中心协调器依据目前网络总体能量状况控制 α, β 值的设定。中心协调器可以通过计算偏低区节点的比例对 α, β 值进行改变,控制多数节点处于充足区或偏低区,以防止出现大量寻路失败或数据拥塞。

3.2 簇树拓扑结构的改进

在 ZigBee 网络中,数据在使用簇树结构进行转发时沿树

型推进,通常到达源节点和目标节点(或源节点的邻居节点和目标节点)的深度最高公共父节点后,再逐层向下传送至目标节点^[6]。数据转发跳数与节点在簇树结构中的深度有关。

如图 1(a)所示,节点 S 的深度为 2, D 的深度为 3,节点 S 要发送数据到节点 D 需要经过 5 跳。如果 D 是中心协调器的子节点,其深度可以降低为 1,如图 1(b)所示,这时 S 发送到 D 的数据经过 3 跳就可以到达。如果 D 节点有子孙节点,则 S 到 D 的所有子孙节点的路径都会缩短 2 跳。在大多数情况下,减小节点在簇树结构中的深度可以有效减少数据沿簇树结构转发的跳数。

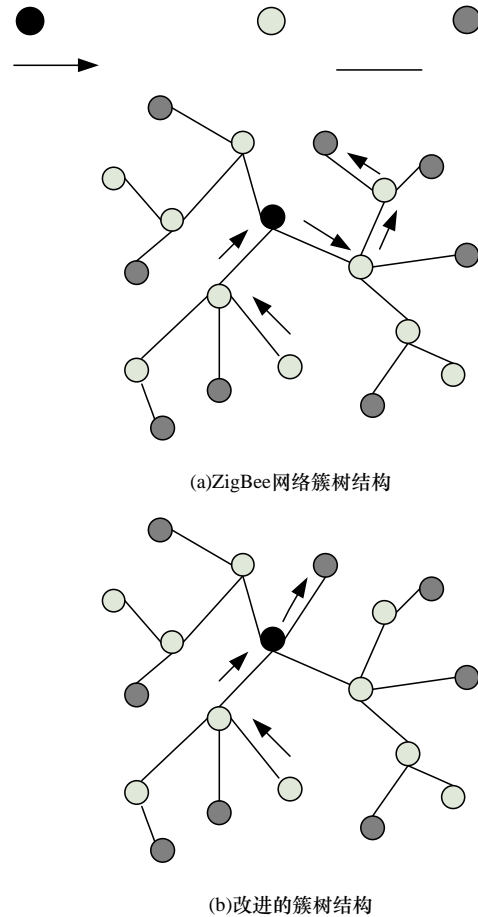


图 1 ZigBee 网络簇路由结构

新节点加入 ZigBee 网络时,通常将第一个发来回应包的路由器节点作为自己的父节点,不考虑路由器节点在树结构中的位置、能量状况和负载情况,往往无法优化路由,并容易造成负载的不均衡。针对这些问题,本文综合考虑路由器的深度、能量区和负载等评价价值对簇树拓扑结构进行优化。优化过程如下:

当新节点申请加入网络时,它首先向网络中的节点发送关联请求,关联请求中含有节点类型。收到关联请求的路由器节点如果能够接受此节点作为自己的子节点,则回复消息 RMSG。RMSG 中含有该节点 16 bit 网络地址、节点深度 d 、能量区域标志以及其当前负载评价量 γ 。 γ 由路由器根据本地信息计算,计算公式如下:

$$\gamma = C_d + RT_c$$

其中, C_d 为节点的子孙节点数目,新节点加入时,沿簇树结构层层向上汇报消息,直到中心协调器,沿途路由器节点将 C_d 值加 1; RT_c 是路由器节点本地路由表中的表项数目。在

对负载量 γ 的设定上,认为节点的子孙节点数目和路由表项数目能从一定程度上反映节点潜在的数据转发任务强度,从而体现路由器的负载度。

新节点对父节点的选择过程如图 2 所示, d_{min} 和 d_{max} 分别为邻居路由器中的最小深度和最大深度。图中依据节点的深度和能量区建立二维坐标系,将邻居路由器分类到不同的类别区域中。新节点对父节点的选择顺序如图中箭头所示,选择到每一类时,如果存在此类节点,则选择这类节点中 γ 值最小的作为其父亲节点。由于邻居节点一定会落在某一个分类中,因此选择过程一定会成功。

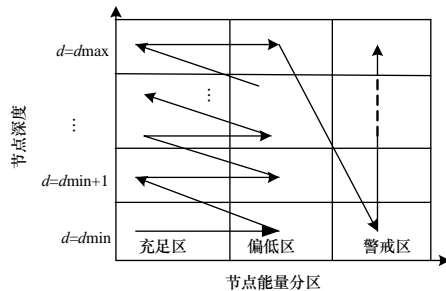


图 2 父节点选择方法

3.3 路由策略的改进

本文针对 ZBR 的不足之处对其进行改进。在新策略中,节点首先利用本地信息和簇树结构寻找转发节点,在需要使用 AODVjr 算法寻找路径时,新策略依据本地信息限制 RREQ 分组的转发范围,并改进对 RREQ, RREP 分组的处理方法。

3.3.1 本地资源

除了节点本身的信息以外,新策略还需要使用节点一跳传输范围内邻居节点的信息。所有节点维护邻居列表 Neighbor List(NL)。NL 中记录邻居节点的如下信息:节点标识 ID, 16 bit 网络地址, 能量分区, 簇树结构中深度 d , 邻居父节点 16 bit 网络地址。新节点加入网络时通过接收邻居节点的 Beacon 消息获取这些信息,并在以后的网络运行中通过监听和与邻居节点交互保持 NL 中信息的实时性。

3.3.2 新策略描述

与 ZBR 相同,新策略中 RN-节点收到数据分组后仍然使用簇树算法进行处理。RN+节点收到数据分组后,如果本地路由表中存在到达目的节点的路由信息,则直接按照路由表进行转发;否则,节点不立即发起 AODVjr 路由发现过程,而是采用一种新的路由算法。新的路由算法包括 2 个阶段。

第 1 阶段 寻找转发节点

RN+节点首先充分利用本地信息和簇树结构寻找下跳转发节点。设本节点为 S, 目标节点为 D, 寻找转发节点过程如下:

- (1) S 搜索 NL, 如果 D 是 S 的邻居节点(包括 S 的父节点)或某邻居节点的父节点,则直接将数据转发给这个邻居节点。否则,执行(2)。
- (2) S 依据 2.2 节中介绍的方法,利用 D 的网络地址、S 的网络地址以及 S 的深度查看 D 是否为 S 的子孙节点。如果是,则按照簇树路由算法计算转发节点地址;否则,执行(3)。
- (3) 用与(2)同样的方法查看 D 是否为 S 某邻居路由器节点(包括 S 的父节点)的子孙。如果 D 是 S 的多个邻居节点的子孙节点,则选择 d 最大的节点为转发节点。如果没有符合要求的邻居节点,则执行(4)。
- (4) 用与(2)同样的方法,查看 D 是否为 S 某邻居路由器

节点(包括 S 的父节点)的父节点的子孙。如果存在多个这样的邻居节点,选择 d 最大的节点为转发节点。如果没有符合要求的邻居节点,则执行(5)。

(5) 根据本地信息寻找转发节点失败,启动改进的 AODVjr 算法寻找路径。

寻找转发节点的流程如图 3 所示。在寻找过程中的每一步,如果找到多个符合条件的转发节点,说明这些节点在簇树结构中是祖先与子孙的关系,选择其中深度 d 最大的转发节点可以最大程度减少数据的转发跳数。算法中将 S 的父节点同其他邻居路由器同样对待,简化了寻找过程。

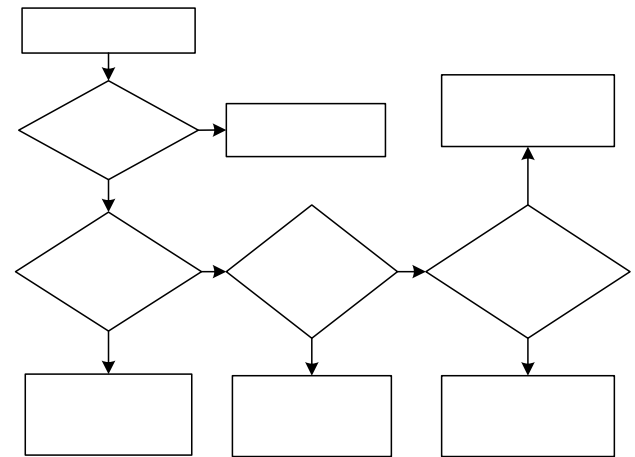


图 3 依据本地信息寻找转发节点

第 2 阶段 改进的 AODVjr 算法

(1) 限制 RREQ 分组的转发范围。首先,改进算法利用簇树结构处理 RREQ 的转发,因此, RREQ 转发长度不应大于簇树结构最大深度(设为 L_m)的 2 倍,将 RREQ 分组的最大跳数设为 $2L_m$,已经过最大跳数个节点的 RREQ 分组将被丢弃。再次,由第 1 阶段寻找转发节点的过程和后面介绍的回复 RREQ 的方法可知,目标节点一定不是发送 RREQ 分组节点的子孙,子点转发来自父节点的 RREQ 分组意义不大,将其丢弃。最后,为了保护低能量节点,处于警戒区的节点不转发任何 RREQ 分组。

(2) RREQ 分组的处理。在 AODVjr 算法中,只能是目标节点来对最先到达的 RREQ 信号做出响应。在改进算法中,当节点接收到同样 RREQ 分组中的第一个时(再次收到相同分组将被丢弃),首先启动第 1 阶段中的方法寻找目标节点的位置。这里,由于接收 RREQ 分组的节点不是目标节点的祖先,可以省略第 1 阶段中的第(2)步。通过第 1 阶段中的寻找,如果目标节点不是当前节点附近节点的子孙,则继续转发 RREQ;如果能够发现目标节点的祖先节点,则当前节点不再转发 RREQ,自己作为目标节点的一个代言节点,立即回复 RREP。代言节点将选择出的转发节点记为下跳节点,并代替目标节点进行定期路由维护。代言节点无须通知路径上的后续节点,下跳节点收到数据包后,将在第 1 阶段算法中立刻发现自己是目标节点的祖先节点,沿簇树结构向下转发数据。

(3) RREP 分组的处理。由于不是只有目标节点才能回复 RREP,可能造成源节点接收到多个 RREP 的情况。在这种情况下,源节点可以保留多个 RREP 作为备选路径,并依据其他信息选择最优路径。本策略在 RREP 分组中加入下面信息:

策略慢。可见，由于新策略在形成簇树结构和进行路由选择
(下转第 118 页)