

无线传感器网络 SMAC 协议的节能改进

苏 俊, 胡访宇

(中国科技大学电子工程与信息科学系, 合肥 230027)

摘 要: 分析 SMAC 协议的工作过程以及自适应侦听的原理, 针对该机制存在的无谓侦听问题提出改进方案。在协议中引入上下节点表, 使节点自行判断是否应当被唤醒, 避免无谓侦听, 减少能量损耗, 同时对该改进方案进行仿真。实验结果表明, 该方案是有效的。

关键词: 无线传感器网络; SMAC 协议; 自适应侦听

Energy-saving Improvement for SMAC Protocol in WSN

SU Jun, HU Fang-yu

(Dept. of Electronic Engineering & Information Science, University of Science & Technology of China, Hefei 230027)

【Abstract】 This paper analyzes the work process of SMAC protocol and the principle of adaptive listening. Aiming at the problem of useless listening existed in this mechanism, an improved scheme is proposed. By introducing the up and down nodes table into this protocol, the nodes can decide whether they are waked up by themselves, which avoids useless listening, and reduces the energy consumption. The scheme is also simulated and experimental results show this improved scheme is effective.

【Key words】 Wireless Sensor Network(WSN); SMAC protocol; adaptive listening

1 概述

无线传感器网络由大量拥有无线通信功能的传感器节点组成, 在环境检测、战场监控、目标追踪和灾害检测等方面都有广泛的应用前景。传感器网络具有低成本、低功耗、布置简单、适应性强等特点, 但由于节点的微型化和低成本化, 在网络中的单个节点所存储的能量非常有限, 且对许多特定的应用不可能随时更换传感器节点的电池, 因此如何延长节点的寿命, 增加网络的使用时间, 成为传感器网络研究的热点, 而对传感器网络 MAC 协议的研究更具实际意义。

根据文献[1]的测量结果, 在无线网络中, 空闲、接收和发送的功率比值为 1:2:2.5, 而文献[2]的测量结果为 1:1.05:1.4, 即节点在空闲时也消耗较多能量。因此, 使传感器节点尽量进入睡眠状态, 减少其空闲侦听的时间在延长其使用寿命中有着重要意义。为有效减少节点能量消耗, 满足传感器网络的特点和需求, 文献[3]提出 SMAC(Self-organizing Medium Access Control for Sensor Networks)协议。本文介绍了 SMAC 的原理, 并针对 SMAC 的特点利用跨层方法提出改进方案, 同时对该方案进行仿真分析。

2 SMAC 简介

SMAC^[3]是针对无线传感器网络设计的 MAC 协议, 其目的是为了减少网络能量损耗。SMAC 使用与 802.11MAC 类似的数据收发机制, 并在此基础上使用周期性的侦听睡眠机制, 使得没有数据发送的节点进入休眠状态, 达到节省能量的效果。同时 SMAC 也有着良好的可扩展性和碰撞避免机制。

通常造成能量损耗的原因主要有碰撞、串听和空闲侦听。SMAC 针对以上各种原因提出相应的解决方法^[4]。

(1)SMAC 使节点都进行周期性的侦听和睡眠, 如图 1 所示。每个节点都周期性地睡眠一段时间, 然后醒来进行侦听, 看是否有其他节点需要通信, 如果没有通信或者在通信结束后节点将重新回到睡眠状态。占空比 $D=T_l/T_f$ 通常能小于

10%, T_f 为周期长度, T_l 为侦听时间, 不但减少了节点空闲侦听的时间, 还能降低串听概率。

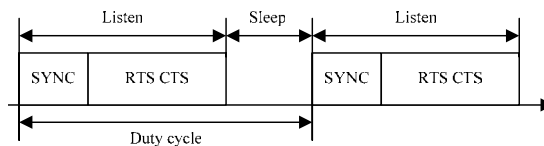


图 1 SMAC 睡眠侦听周期

(2)SMAC 使用与 802.11 类似的 RTS/CTS 机制。节点在发送数据前将进行载波侦听, 如果竞争信道失败它将重新进入睡眠直到信道空闲。在发送节点确认可以发送数据后, 它将向接收节点发送一个 RTS 数据包, 接收者收到 RTS 以后将发送 CTS 应答, 收到 CTS 后发送者就能向接收者发送数据信息包, 当接收者完全接收数据信息包后返回 ACK 确认, 发送者收到 ACK 后表示此次通信结束。通过 RTS/CTS 机制, 减少了碰撞和串扰发生的概率。

(3)为有效地与邻居节点的侦听睡眠周期匹配, SMAC 还使用局部同步机制。在节点启动时, 先会侦听一段时间, 如果没有收到来自其他节点的 SYNC 包, 就自行产生自己的侦听周期并向周围广播。如果节点收到来自邻居的 SYNC 包并且自己尚未确定侦听周期, 就将该侦听周期作为自己的侦听周期, 并将在下一个周期的侦听时间广播 SYNC 包。如果节点在广播自己的 SYNC 包后收到来自其他节点的 SYNC, 并且没有发现其他邻居, 就将收到的 SYNC 包所指示的侦听周期作为自己的侦听周期; 若有其他邻居, 就同时使用 2 个侦听周期。通过局部调度同步算法, 确保邻居节点在单跳内实现同步, 避免了碰撞和串扰的发生。

作者简介: 苏 俊(1982 -), 男, 硕士研究生, 主研方向: MANET, 无线传感器网络; 胡访宇, 教授

收稿日期: 2008-09-30 **E-mail:** suj@mail.ustc.edu.cn

3 SMAC 的改进

虽然周期性的睡眠侦听减少了系统的空闲侦听,但同时造成时延增加。因此,SMAC 提出了自适应侦听方法^[4],如图 2 所示。

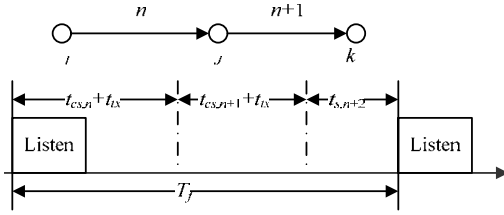


图 2 SMAC 自适应侦听过程

自适应侦听的基本思想是：当某一节点仅能听到邻居节点发送的 CTS 包时,且自己不是目的节点时,该节点将进入睡眠,直到此次传输结束后醒来。如果此次传输的下一跳节点为发送节点的邻居节点时,它将收到 RTS 包;如果它只是接收节点的邻居时,它将会收到 CTS 包。因此,醒来的节点都只会影响到发送者和接收者其中之一的通信,且它们都能从 RTS 或 CTS 中的持续时间域中获知当前通信的持续时间。因此,收到数据的节点就能立即将数据传给其下一跳,而不用等待另一个调度周期。而其他节点在等待一段时间后如果没有听到任何数据包,则将继续睡眠直到下一个调度周期。

假定载波侦听造成的延时为 t_{cs} ,由于数据包长度固定,因此设传输延时固定,设为 t_{tx} 。在调度周期 T_f 大于 $t_{cs,n} + t_{cs,n+1} + 2t_{tx}$ 时,数据包将可以在一个调度周期中传递 2 跳,因此,对于经过 N 跳的数据包平均时延为^[3]

$$D(N) = t_{s,1} + (N/2 - 1)T_f + t_{cs,N-1} + t_{cs,N} + 2t_{tx}$$

而如果没有自适应侦听,则其时延为

$$D(N) = t_{s,1} + (N - 1)T_f + t_{cs,N} + t_{tx}$$

因此,自适应侦听机制有效减少了系统时延。但在自适应侦听唤醒的节点中,大多数节点与本次通信无关,这类节点称为额外节点。而由于这些节点都需要等待一段时间才能返回睡眠,因此增加了能量损耗。为简化问题,假设某节点对通信节点周围含有 N 个只能收到 CTS 的邻居节点,每个节点侦听功率为 P_l ,发送功率为 P_t ,睡眠功率为 P_s 。假设发送单个数据包的时间为 T_{data} ,发送单个控制包的时间为 T_{ctl} ,且 $t_{tx} = T_{data} + 3T_{ctl}$,而 T_{data}/T_{ctl} 表示信息数据包长度与控制数据包长度的比值,则在一个数据发送周期中,所消耗的能量为发送数据包、空闲侦听以及睡眠消耗能量的总和,即

$$e_{total} = P_l t_{cs,n} + 2P_t t_{tx} + P_l t_{cs,n+1} + 2NP_l t_{wait} + P_s (T_f - t_{tx} - t_{cs,n}) + P_s (T_f - 2t_{tx} - t_{cs,n} - t_{cs,n+1}) + P_s (T_f - t_{cs,n+1} - t_{tx}) + (N-1)P_s (T_f - 2t_{wait})$$

如果只考虑在自适应侦听中进行通信的节点,则总共消耗的能量为

$$e_{useful} = P_l t_{cs,n} + 2P_t t_{tx} + P_l t_{cs,n+1} + 2P_l t_{wait} + P_s (T_f - t_{tx} - t_{cs,n}) + P_s (T_f - 2t_{tx} - t_{cs,n} - t_{cs,n+1}) + P_s (T_f - t_{cs,n+1} - t_{tx})$$

因此,设额外消耗的能量均值与总能量均值的比值为 η ,称为额外能量系数。若使用表 1 中的参数,则该系数与 N 的关系如图 3 所示。这里, $\eta = 1 - \frac{E[e_{useful}]}{E[e_{total}]}$,且由于 $P_s \ll P_l$,

占空比为 10%,因此由睡眠造成的能量损耗对 η 的影响可以忽略不计。

表 1 仿真主要参数设置

仿真参数	参数值
数据包长度	100 Byte
控制包长度	10 Byte
发射功耗	24.75 mW
接收功耗	13.5 mW
空闲功耗	13.5 mW
睡眠功耗	0.015 mW
场景尺寸	1 500×1 500
节点数	100
模拟时间	400 s
占空比	10%

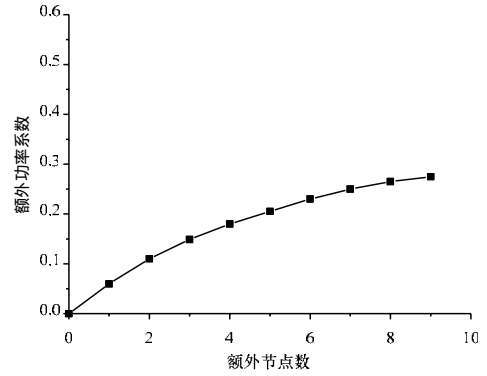


图 3 额外能量系数曲线

从图 3 可以看出,随着额外节点数的增多,额外能量系数也随之增大,因此,避免额外唤醒能够有效减少系统能量损耗。

在传感器网络中,数据经传感器节点向一个或多个基站发送。当基站固定时,拓扑结构相对稳定,使得节点路由表也相对稳定。通常情况下只有在下一跳能量过低或者失效时节点才改变其路由表。因此,经由某一节点转发的数据的下一跳节点在较长的时间内是稳定的,这使得节点通过 RTS 或 CTS 来获知自己是否应该醒来成为可能。基于上述考虑,笔者在 MAC 层维护一张上下节点表,表中记录该节点上一跳和下一跳节点的 MAC 地址。这里,一个节点可以有多个上一跳和下一跳节点。在更新路由表的同时更新上下节点表,在收到 RTS 或 CTS 包时,节点在表中查找,只有当自身为节点的上一跳或者下一跳才会醒来,否则继续睡眠。使用 ns-2 软件对改进后的 SMAC 协议进行仿真,所有节点将 SMAC 协议作为 MAC 层协议,将 DSR 协议作为路由协议。场景中节点位置随机,且在模拟过程中保持节点位置不变。仿真结果如图 4 所示。

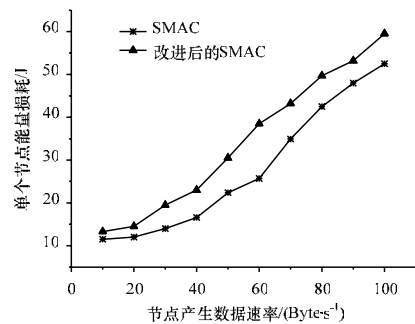


图 4 平均节点能耗比较

(下转第 121 页)