

无线传感器网络操作系统调度策略

尹震宇, 赵海, 林恺, 刘楠, 徐久强

(东北大学信息科学与工程学院, 沈阳 110004)

摘要:提出了一种在无线传感器网络操作系统中可以同时针对周期性任务和非周期性任务进行抢占式调度操作的EF-RM调度策略。在无线传感器节点上执行的任务负载较重的情况下,该调度策略可以保证重要任务的优先执行,此外当无线传感器节点空闲时,通过将节点带入睡眠状态,实现无线传感器节点的节能。所提出的任务调度策略在TinyOS上进行实现,并通过实验测试得出,在总能耗代价增加较少的情况下可以有效地提高系统在较重负载情况下的响应性能。

关键词: TinyOS; RM调度策略; 无线传感器网络; 操作系统

Scheduling Strategy on Wireless Sensor Networks Operating System

YIN Zhen-yu, ZHAO Hai, LIN Kai, LIU Nan, XU Jiu-qiang

(School of Information Science and Engineering, Northeastern University, Shenyang 110004)

【Abstract】 This paper proposes the preemptive scheduling strategy, EF-RM for wireless sensor network operating system. EF-RM can archive the task scheduling for periodical task and un-periodical task. By using the EF-RM scheduling strategy, the important task can be executed with high priority. When a wireless sensor node is idle, the EF-RM can bring the node into sleep mode to save the energy. The scheduling strategy is achieved in the TinyOS. And the experiment shows that EF-RM scheduling strategy can improve the response of the wireless sensor nodes in heavy load with a little cost of energy.

【Key words】 TinyOS; RM scheduling strategy; wireless sensor networks; operating system

无线传感器网络操作系统由于受到处理器资源、存储器空间以及能源等方面的严格限制,要求其占用的存储器空间更小,同时在任务调度管理效率及安全性等方面要求更高^[1]。而在现有的无线传感器网络操作系统中,最常采用的是先来先服务(first come first service, FCFS)调度策略,该策略无法保证无线传感器节点在任务负载较重情况下重要任务的优先执行。无线传感器网络在实际工作中,节点上可能同时运行着一个或者多个周期性执行的传感采集任务,同时,在无线传感器节点上,还可能会突发性地启动诸如维护无线网络通信或者调整自身配置信息的系统任务,如果无法保证这些系统任务的优先运行,将会导致节点工作状态异常甚至会引起无线传感器网络整体瘫痪。而FCFS任务调度策略显然是无法保证无线传感器节点在任务负载较重情况下重要任务的优先可靠执行。

本文提出了一种基于RM(rate monotonic)调度策略的紧急任务优先任务调度策略(emergency task first rate monotonic, EF-RM)。该调度策略保证了无线传感器节点中实时性任务以及重要任务的优先可靠运行,同时在无线传感器节点上增加了实现系统休眠的任务,保证无线传感器节点在任务队列中没有其它需要执行的任务时可通过执行系统休眠任务来减少无线传感器节点的能耗。

1 相关工作

RM^[2]调度策略是由C L Liu和J W Layland于1973年提出的,该算法根据任务的执行周期来分配固定的优先级,执行周期越短的任务分配的优先级越高^[3]。假设任务 τ_i 的执行周期为 T_i ,最坏执行时间为 C_i ,任务执行的截止期限为 D_i ,相位为

I_i ($0 \leq I_i < T_i$), 则可以得到任务 τ_i 的执行期为从 I_i+kT_i 到 $I_i+(k+1)T_i$ 的一段时间。假设有 n 个任务构成任务集 $\Gamma=\{\tau_1, \tau_2, \dots, \tau_n\}$,任务集合 Γ 中的任务已经按照任务的执行周期由小到大进行排序,即有 τ_1 到 τ_n 的执行周期为 $T_1 \leq T_2 \leq \dots \leq T_n$,则任务 τ_i 在 I_i+kT_i 到 $I_i+(k+1)T_i$ 时间段内拥有优先权 i ($1 \leq i \leq n$),并且可以抢占低于该优先级别的其它任务。当任务集 Γ 中任务 τ_i 符合 $C_i \leq D_i \leq T_i$,任务集 Γ 中的 n 个任务的处理器利用率 U 为

$$U = \sum_{i=1}^n \frac{C_i}{T_i} \quad (1)$$

并且当 n 个任务的处理器利用率符合 $U \leq n(2^{1/n} - 1)$ 时,可以保证所有的任务都能在其截止期前完成。RM调度策略是最优静态优先级抢占式调度策略^[4]。

2 EF-RM 调度策略在 TinyOS 上的实现

2.1 EF-RM 调度策略中任务优先级的设计

无线传感器节点上运行的任务可以分为2类:(1)周期性任务,例如节点上运行的传感器数据采集任务,并且在很多应用环境中,节点的任务队列中同时包含多个传感器数据采集任务。(2)突发性任务,主要包括进行系统相关维护配置操作或者进行无线通信相关操作的系统任务,该类任务的启动

基金项目: 国家“863”计划基金资助项目(2001AA415320);国家自然科学基金资助项目(69873007)

作者简介: 尹震宇(1979-),男,博士研究生,主研方向:无线传感器网络,Ad Hoc网络;赵海,教授、博士生导师;林恺,博士研究生;刘楠,硕士研究生;徐久强,教授

收稿日期: 2006-10-08 **E-mail:** cmy@neuera.com

没有周期性规律,和无线传感器节点上运行的其它任务相比,该类任务对系统的响应性要求较严格,如果该类任务不能在截止时限内执行完成,将对无线传感器节点自身的工作状态以及整个无线传感器的正常工作造成较严重的影响。由于上述原因,在 EF-RM 中将所有突发性任务的优先级设定为高于周期性任务的优先级,即非周期性的突发启动的任务可以中断当前的所有的周期性任务,而周期性任务的优先级则依靠任务的执行周期确定。

2.2 任务列表数据结构设计

EF-RM 调度策略由于进行任务抢占操作,因此需要在原始的 TinyOS 的任务列表数据结构中添加多项纪录任务相关状态的属性,修改后的任务列表数据结构如下:

```
typedef struct
{
    void (*tp) ();
    uint8_t status;
    uint16_t dead_line;
    uint16_t period;
    uint16_t phase;
    uint16_t run_time;
    uint16_t idle_time;
    uint8_t priority;
    uint16_t stack_point;
} TOSH_sched_entry;
```

该数据结构中,*tp 指针为任务的入口地址。变量 status 记录的是任务的执行状态,对于周期性任务,当任务被激活时该变量被设置为 1,表示该任务处于就绪状态;当任务获得处理器使用权时该变量被设置为 2,表示该任务正在执行;当任务在执行过程中被高优先级的任务中断时该变量被设置为 3,表示该任务处于挂起状态;当任务在当前任务周期内执行完成时该变量被设置为 4,表示任务处于完成状态。对于突发性任务,由于在任务执行结束时操作系统将回收任务占用的系统存储器空间以及任务列表记录,因此对于 status 变量只使用到就绪状态、执行状态以及挂起状态。period 记录的是周期性任务的执行周期。phase 表示周期性任务的启动相位。run_time 表示任务运行时间,对于周期性任务,该变量记录的是在一个任务周期内处于执行状态的累计时间,对于突发性任务,则是任务处于执行状态的总时间。idle_time 对于周期性任务,表示的是其在每个周期内处于就绪状态和挂起状态的累计时间,而对于突发性任务,则记录的是任务处于就绪状态和挂起状态的总时间数。dead_line 对于周期性任务表示从每个执行周期开始距离任务执行结束所允许的最大时限,对于非突发性任务,则表示从任务启动开始距离任务执行结束所允许的最大时限。priority 记录的是突发性任务的优先级,规定数值越小表示优先级越高,由于在 TinyOS 的任务队列中最大支持 8 个任务,因此对于突发性任务可以使用的优先级取值范围为 0~7,此外对于周期性任务,该变量取值为 254,对于系统睡眠任务,该变量取值为 255。变量 stack_point 中存储的是任务堆栈指针。在使用 EF-RM 调度策略时,需要将 TinyOS 的共享堆栈空间改为每任务独享的独立堆栈空间,在本设计中划分出 8 个 256B 的独立的堆栈空间,加上操作系统和无线通信协议栈占用的大约 2.2KB 内存空间,总内存使用量小于 4KB,可以保证在现有的无线传感器节点上运行。此外在设计中涉及到的有关时间的变量均使用系统时钟计时个数为计时单位以便于代码设计。

2.3 EF-RM 任务调度函数的设计实现

图 1 为在 TinyOS 上实现的 EF-RM 调度策略的任务提交函数 TOS_post 的程序流程图。

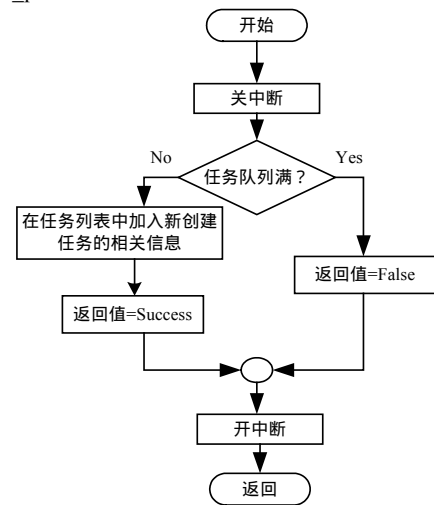


图 1 TOS_post 程序流程

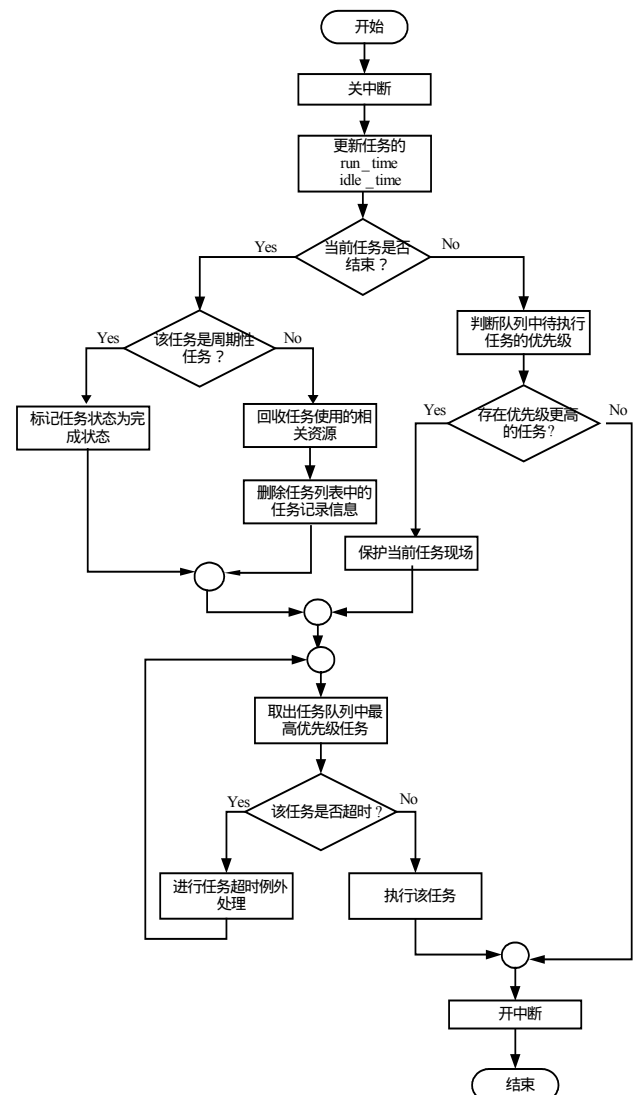


图 2 TOSH_run_task 程序流程

TOS_post 的函数原形如下: bool TOS_post (void (* tp) (), uint16_t priority, uint16_t period, uint16_t dead_line, uint16_t phase)。所有新任务都通过该函数提交到 TinyOS 的

任务队列中。EF-RM 调度函数 TOSH_run_task 的程序流程如图 2 所示。

对于任务的优先级,首先判断变量 priority,规定数值越小优先级越高。突发性任务的优先级为 0 到 7,远高于周期性任务,可以保证突发性任务优先调度处理,而周期性任务其优先级都为 254,此时依靠其任务周期进行优先级排序,任务周期越小优先级越高。此外在任务队列的末尾额外添加了一个系统休眠任务 idle_task,该任务不占用任何用户任务空间及资源,它的功能是将无线传感器节点设置为睡眠状态,直到通过无线通信中断或时钟中断等将节点唤醒,该任务在系统初始化时被永久设成最低优先级 255,该任务可以被任意其它任务中断。设置该任务可以解决系统处于空闲时的节能问题,同时由于该任务的存在,任务列表永远不为空,可以简化任务调度函数的设计。此外规定系统在下列情况时才进行任务抢占调度:新任务提交、周期性任务在每个任务周期被启动、突发性任务执行完成、周期性任务在当前周期内执行完成以及系统响应中断的时候,当上述任意一种情况发生时,系统将启动任务调度函数 TOSH_run_task 进行抢占调度,如果任务队列中存在更高优先级的任务,当前任务将被中断,系统将运行队列中的最高优先级的任务。每当从任务队列中取出一个待执行任务时,首先要判断该任务是否超时,判断条件为:run_time+idle_time≤dead_line,对于不符合该条件的周期性任务将在当前周期内被放弃执行,而突发性任务则将被丢弃。

3 性能测试与评价

EF-RM 调度策略在 TinyOS v1.1.1 上实现,并使用基于 FCFS 调度策略的 TinyOS 进行比较。实验包括两部分,使用的测试平台分别为实际搭建的无线传感器网络环境及 PowerTossim 仿真软件。

(1)实验的测试环境为 MPR2400 无线传感器节点构建的实际无线传感器网络。网络中的普通节点使用 2 节 1200mAh 工业级 AA 碱性干电池作为工作电源。基站节点由一个 MIB510 编程下载板以及一个 MPR2400 无线传感器节点构成,使用外接电源供电。该实验环境如图 3 所示,包括 2 个普通无线传感器节点 A、B,以及 1 个基站节点 C。

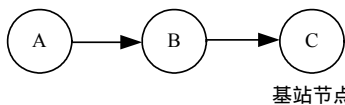


图 3 实验环境示意图

实验中,将 3 个节点放置在开阔空旷的场地中,并将 3 个节点沿直线排列开,使节点 A 的通信范围可以覆盖节点 B 但是不能够覆盖节点 C,而节点 B 的通信范围可以同时覆盖 A 和 C。节点 B 分别运行采用 FCFS 调度策略以及 EF-RM 调度策略的 TinyOS 操作系统。节点 B 上作为突发性任务的是多跳路由转发任务,执行时限设置为 900ms。此外还包含多个执行连续加法求和运算操作的周期性的测试任务,通过设定这些周期性任务的连续加法运算量可以很方便地控制它们的总执行时间。周期性任务具体情况如表 1 所示。

为了简化测试,当使用 EF-RM 调度策略时,这些周期性任务的周期均设置为 1s,启动相位均设置为 0。当使用 FCFS 调度策略时,测试任务 T₀~T₄ 的执行内容和执行时间按照表 1 给出的参数设置,并且这些测试任务的调用周期也设为 1s。规定在每个周期开始的时候,记录下任务列表中未执行完的

任务,清空任务列表并在任务列表中重新放入所有待执行的用户任务。节点 A 运行的是专门编写的数据包发送任务,模拟邻居节点向节点 B 发送数据包。在 A、B、C 3 个节点上运行的 TinyOS 中加入了时间同步模块,如果需要节点 A 发送无线通信数据包,则规定 A 节点在 B 节点的每个任务周期开始的时间点上向后延时 100ms,然后连续发送 5 个数据包。

表 1 待执行周期性任务

任务名称	任务执行时间/ms	任务周期/s	执行时限/s	相位
T ₀	50	1	1	0
T ₁	100	1	1	0
T ₂	150	1	1	0
T ₃	200	1	1	0
T ₄	500	1	1	0

在每个用户任务周期内,节点 A 发送的无线通信数据,如果没有被节点 B 转发给节点 C,则判定该数据包超时,会在下一周期开始时直接被丢弃。每个任务周期内,节点 B 和节点 C 的运行情况如表 2、表 3 所示。

表 2 FCFS 任务调度运行情况

B 节点用户任务的启动顺序	C 节点接收数据包数/pkgs	备注
A 节点发送数据包		
T ₄	4	发生数据丢包
T ₃ ,T ₂ ,T ₁ ,T ₀	4	发生数据丢包
T ₄ ,T ₃ ,T ₂ ,T ₁ ,T ₀	0	数据丢包,T ₀ 超时
A 节点未发送任何数据		
T ₄ ,T ₃ ,T ₂ ,T ₁ ,T ₀	0	T ₀ 超时

表 3 EF-RM 任务调度运行情况

B 节点用户任务的启动顺序	C 节点接收数据包数/pkgs	备注
A 节点发送数据包		
T ₄	5	T ₄ 超时
T ₃ ,T ₂ ,T ₁ ,T ₀	5	T ₃ 超时
T ₄ ,T ₃ ,T ₂ ,T ₁ ,T ₀	5	T ₃ ,T ₄ 超时
A 节点未发送任何数据		
T ₄ ,T ₃ ,T ₂ ,T ₁ ,T ₀	0	T ₄ 超时

理想情况下节点 B 转发一个无线通信数据包耗时约 100ms。由于任务调度本身需要一定的时间开销,因此在每个任务周期内,可用的任务净运行时间要小于 1s。由表 2 和表 3 可以看出,FCFS 调度策略中任务优先级完全依靠任务进入队列的先后顺序而确定,无法保证重要任务的优先执行。而在 EF-RM 调度策略中操作系统对高优先级任务实现抢占式调度,保证了重要任务的优先执行。

(2)实验使用 PowerTossim 进行仿真测试。使用的是 TinyOS v1.11 自带的 PowerTossim 软件。

该实验中每个节点上运行的用户任务为 TinyOS 自带的例子程序 Surge,在使用 EF-RM 调度策略的 TinyOS 中对 Surge 进行了修改,确保其可以正常运行。仿真过程中,节点以 8s 为周期发送采集到的传感数据,无线通路为理想情况,不需要考虑无线信号、误码率等问题导致的丢包现象。但当无线传感器节点分布密度较高时,对于连接度很高的多跳路由转发节点由于需要同时执行传感数据采集任务以及多跳路由转发任务,任务负载很重,可能会发生过载现象。表 4、表 5 为无线传感器网络运行 1 000s 后的实验结果。

在本实验中,丢包率计算公式为

1 - (基站节点收到的传感信息数据包总数/无线传感器网络中所有节点发送的传感信息数据包总数)

由表 4 以及表 5 的数据可以得出,采用 EF-RM 调度策略 (下转第 82 页)