

油罐车在途状态监测系统及其任务调度策略

惠 飞,赵祥模,杨 飞

(长安大学 信息工程学院,陕西 西安 710064)

摘 要:为了实时监测油罐车在行驶途中的温湿度、车体姿势、支撑座压力、进出油口是否漏油等安全信息,以超低功耗的 ATmega1281 微控制器和 AT86RF230 射频模块为核心,设计了感知油罐车安全信息的无线传感器网络节点,构建了多参数在途监测系统。针对无线传感器网络操作系统 TinyOS 的 FCFS 调度策略对实时性强的任务响应慢及吞吐量低等缺点,提出了一种基于多队列任务调度策略。研究表明:采用实时任务和优先级任务分级调度,可以设置动态优先级和任务超时中断,满足了系统对高实时性和高优先级任务的响应,并保证低优先级的任务也能在适当时间内执行;仿真及实际测试表明,设计的油罐车在途状态监测系统在平均能耗增加不大于 3% 的情况下保证了监测系统的实时性。

关键词:交通信息与控制工程;无线传感器网络;油罐车;TinyOS 操作系统;多队列任务调度

中图分类号:TP313

文献标志码:A

Design of tanker sensor node based on WSN and improvement of TinyOS tasks scheduling

HUI Fei, ZHAO Xiang-mo, YANG Fei

(School of Information Engineering, Chang'an University, Xi'an 710064, Shaanxi, China)

Abstract: For real-time monitoring of temperature, humidity, car body posture, carriage pressure, inlet and outlet mouths oil leaks and other security information while driving oil tank truck, the wireless network node of monitoring tanker security information was designed with ultra-low power of ATmega1281 microcontrollers and AT86RF230 RF modules. To resolve the problem of TinyOS FCFS scheduling strategy which not able to give strong real-time tasks timely response and low message treatment, a multi-task queue scheduling strategy is proposed, which uses grading scheduling for real-time tasks and priority tasks and sets dynamic priority and task overtime interrupts. The algorithm can meet the system's response of real-time and high-priority tasks, make low-priority tasks in appropriate within the time to get scheduling opportunity. The TinyOS test result indicates that the tanker sensor node can realize the real-time monitoring within the 3% increasing of power consumption. 3 tabs, 14 figs, 9 refs.

Key words: traffic information and control engineering; WSN; tanker; TinyOS; multi-queue task scheduling strategy

0 引言

无线传感器网络(WSN, wireless sensor networks)是一种全新的信息获取平台,能够实时监测和采集网络分布区域内的各种监测对象的信息,并将这些信息发送到网关节点,以实现复杂的指定范围内的目标监测与跟踪,具有快速展开、抗毁性强等特点^[1]。无线传感器网络节点使用 TinyOS 操作系统, TinyOS 在任务调度上采用非剥夺的先来先服务(FCFS, first-come-first-served)调度策略,这样的设计将有利于减少系统对存储空间的需求^[2-3]。然而,这种系统不能对实时性很强的网络任务做出及时的响应,并且还会出现任务丢失、通信吞吐量低等情况^[4]。

目前,针对 TinyOS 调度策略的不足已经做了大量的研究和改进,文献[5]提出的双环调度策略,将单循环队列改为具有不同优先级别的双循环队列,并为不同任务分级占用 2 个堆栈,根据队列中任务时限的变化,动态调整两队列间的任务切换,以最大程度地保证任务完成,提高了系统的响应速度,但吞吐量较低;文献[6]提出一种基于优先级的软实时任务调度策略,可以有效提高系统的吞吐量,但不能满足系统实时性的要求;文献[7]提出的 EF-RM 调度策略,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。文献[8]提出的 IS-EDF 调度策略,通过动态调整执行任务的优先级,保证重要任务的实时性;文献[9]提出的 HRRF 调度策略,采用对于实时性较强的任务优先级调度策略,满足对实时任务的响应。

以上几种调度算法都是高级的可剥夺的调度策略,很好地改善了系统对实时性的要求,提高了系统的响应速度和吞吐量,但是这些改进算法需要大量额外的系统资源,而且需要考虑资源的共享和死锁问题,加大了无线传感器节点的负担,因此,在资源受限的无线传感器网络节点上使用并不是一个很实用的办法。为此,本文针对油罐车监控,提出了一种基于无线传感网络的油罐车在途状态监测系统。针对该类应用中对感知节点实时性要求较高的特性,设计了一种实用的多队列任务调度策略,既满足了节点对实时任务和高优先级任务的及时响应,又提高了系统对所有任务的吞吐量,并有效防止节点进入阻塞或死循环等状态。

1 基于无线传感器网络的油罐车在途监测系统设计

1.1 传感器节点硬件设计

无线传感器网络节点通常是由传感器模块、处理器模块、无线射频模块和能量模块组成,结构如图 1 所示。

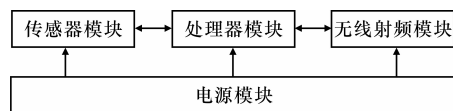


图 1 无线传感器网络节点结构

节点硬件中传感器模块主要负责信息的采集,并将其他各种信号转变成数字信号传给微处理器,微处理器模块负责整个传感器网络节点的操作,存储和处理自身采集的数据,以及其他节点发送来的数据,实现网络安全可靠的通讯协议。无线射频模块负责与其他网络节点进行无线通信,交换控制信息和收发采集数据。电源模块负责提供节点工作的电源,通常采用 AA 电池,以便减小节点的体积。

无线传感器网络节点选用 ATmega1281 微处理器芯片和 AT86RF230 射频芯片,构成嵌入式小型无线测量系统。ATmega1281 是一款 8 位 AVR 低功耗微处理器,内置拥有 128 kB 的可编程 Flash, 4 kB 的 EEPROM, 8 kB 的 RAM, 54 个通用 I/O 引脚, 2 个 USART 串口以及 8 个 10 位 ADC 通道,也内置有 JTAG 接口,同时,为了减少能耗,支持 6 种睡眠模式。AT86RF230 是一款专为 IEEE802.15.4 和 ZigBee 协议应用开发的高性能 RF-CMOS 2.4 GHz 无线收发器,具有超低功耗及超低供电电压的特性,节点的硬件连接及节点实物如图 2、下页图 3 所示。

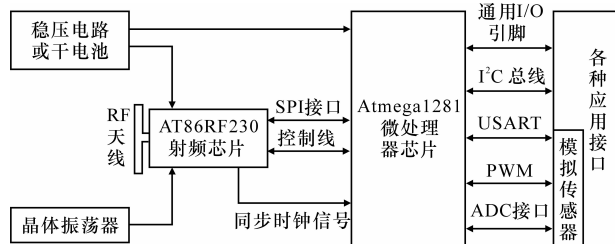


图 2 无线传感器网络节点硬件连接

1.2 无线传感网络节点的部署及网络路由方式

油罐车在行驶途中需要实时监测车辆的状态信息,常用到温度传感器、气体传感器、压力传感器和陀螺仪。根据油罐车的结构,需要在加油口、放油口和放油管道处放置气体传感器,监测在行驶途中是

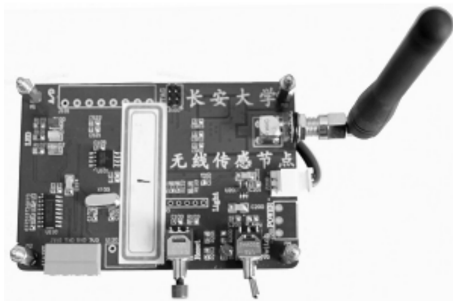


图 3 无线传感网络节点实物

否有泄漏的情况发生。温度传感器放置在车体四周,监测油罐车的车表温度。在车体的支承座处放置压力传感器,监测油罐与底座之间的连接状态。陀螺仪用来监测车辆在行驶途中的状态。整个系统的节点部署如图 4 所示。

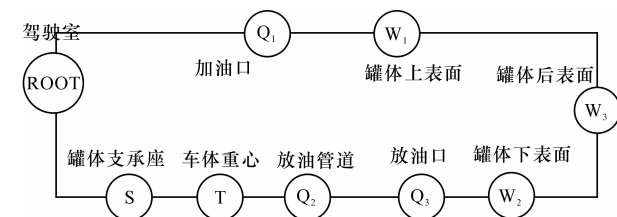


图 4 油罐车节点部署

图 4 中: W_1 、 W_2 、 W_3 为温度传感器节点; Q_1 、 Q_2 、 Q_3 为气体传感器节点; T 为陀螺仪传感器节点; S 为压力传感器节点; $ROOT$ 为根节点。

无线传感器网络的路由协议负责寻找源节点和目的节点间的优化路径,并沿此优化路径正确转发数据包。因为传感器节点能量有限、动态拓扑结构和数据融合处理等特点,其路由协议设计在考虑到功能实现的同时,还需兼顾到能量有限和能量均衡等问题,以提高整个网络的生存期和鲁棒性。

网络采用层次型拓扑结构,进行网络节点静态分簇和按条件在子簇内周期性选择簇头节点,通过定期判断节点剩余能量来轮换选取簇头。簇头负责簇内数据融合与预判,并将整合信息发送到根节点,整个网络形成一个以簇为基本单位的树形拓扑,如图 5 所示。

簇的建立过程可以分成 3 个阶段:簇头节点的选择、向根节点通知新的簇头节点和调度机制。传感器节点的路由如图 6 所示。

图 6 中: W_1 、 W_2 、 W_3 构成一个静态簇; Q_1 、 Q_2 、 Q_3 构成另一个静态簇; W 和 Q 分别表示通过簇头选取算法确定的簇头。该路由算法具有以下特点。

(1)减少了传送到 $ROOT$ 节点的信息数量,簇头节点负责融合和预判不同传感器节点的数据,并

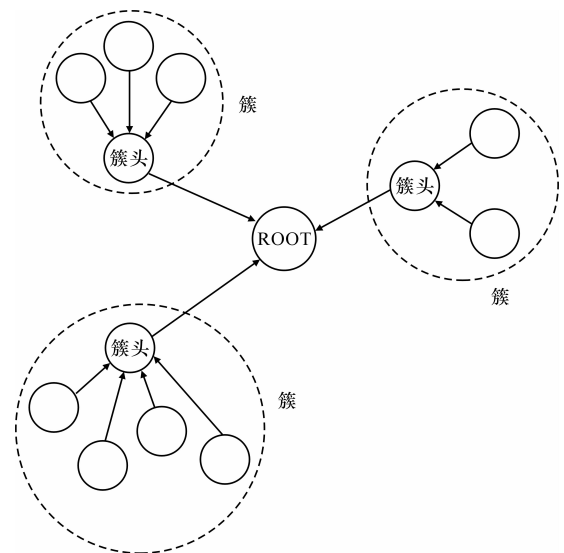


图 5 簇树型网络拓扑

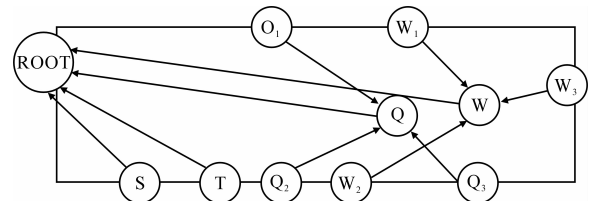


图 6 传感器节点路由

将融合后的数据发送到 $ROOT$ 。

(2)减少了簇内和簇间的冲突。

(3)数据采集是集成的和周期性的,该路由适应于连续监控的应用系统。

(4)簇头节点进行简单的预判,降低非关键数据的传输次数,可以减少传感器节点能量的消耗。

(5)周期性的重选簇头节点,保证无线传感器网络具有统一的能量分布,延长使用时间。

2 节点任务调度策略设计

2.1 任务队列

为了能够及时响应时限要求严格的实时性任务,本文设计的调度器新增了一个实时任务队列 (RT_Queue),用来存放实时任务。该队列的优先级在所有队列中是最高的,采用的 $SEDF$ (simplified earliest deadline first) 调度算法是在 EDF 算法的基础上精简得来的,不会有过高的计算开销。对于非实时任务新增一个动态优先级队列 (DP_Queue),用来存放非实时优先级任务,并设计了一种简易的动态优先级调度算法 LIP (liner incremental priority) 对该任务队列进行调度。为了兼容原有的 $TinyOS$ 任务模型,保留 $TinyOS$ 内核中基本任务队列

(Basic_Queue)。

(1)RT_Queue 队列存放实时任务。调度器会参考任务截止时间,对任务队列顺序做出相应调整,该队列中的任务拥有最高运行级别,会被优先调度。

(2)DR_Queue 队列存放动态优先级任务。该任务队列中任务的优先级会动态递增,其运行级别介于 RT_Queue 与 Basic_Queue 之间。

(3)Basic_Queue 队列存放基本任务。该任务队列的总体运行级别处于最低端,任务队列按 FCFS 的调度策略进行调度。

2.2 任务调度策略

2.2.1 实时队列任务调度

pushRTTask(uint8_t id,uint32_t deadline) 函数是 RT_Queue 队列投递函数 TaskRT.postTask[id] 的执行主体。该函数实现了实时任务队列的任务调度策略 SEDF,其核心思想是:通过参考被投递的实时任务相对截止时间来计算绝对时限,并与队列中等待任务的截止时间依次做比较,使得该实时任务队列总能保持较近运行截止时限的任务排在队列前面,以便能够优先调度。实时任务调度策略流程如图 7 所示。

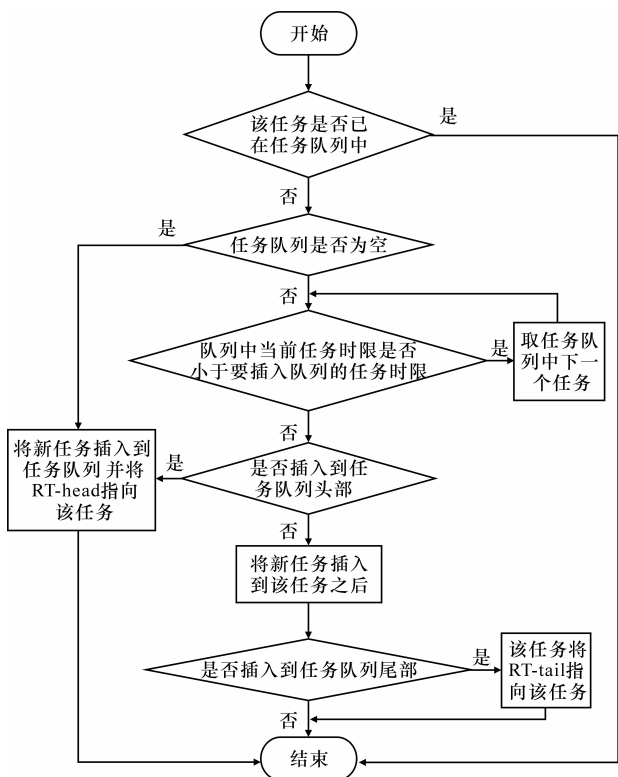


图 7 实时队列任务调度流程

首先检测该任务是否已经在任务队列中,如果有,则不能再将其投入队列;如果一个任务想连续执行的话,可以在投递该任务的组件中的任务代码末

尾处再次投递该任务。RT_time[ID]中存放的是实时任务的绝对限值,每次要插入新的任务时就要在实时任务队列中挨个比较任务的绝对时限值,并通过设置 2 个局部变量 t_{curr} 和 t_{prev} ,将新任务插入到实时任务队列的正确位置上,使这 2 个变量分别指向当前任务 ID 和前一个任务 ID。当任务插入到任务队列头部时,需要用 RT_head 指向该任务;同理,将任务插入到任务队列尾部时,要用 RT_tail 指向该任务。

2.2.2 动态优先级队列任务调度

pushDPTask(uint8_t id,uint8_t priority) 函数是 DP_Queue 队列投递函数 TaskDP.postTask[id] 的执行主体,event Timer.fired() 函数是一个时钟中断时间函数。以上 2 个函数合作实现了动态优先级任务队列的任务调度策略,其核心思想是:任务进入就绪队列之前均有一个初始的优先级,在进入该队列之后,通过定期增加任务队列中等待任务的优先级,以确保先进入队列的任务不会一直被频繁投递的高优先级任务所阻塞,并将后来投递的任务安插到队列的合适处,使得该动态优先级任务队列总能保持较高优先级的任务排在队列前面的规则,以便能够被优先调度。

pushDPTask 函数使得任务队列能够按照给定优先级的大小进行插入队列,其具体算法流程与实时任务队列调度策略大致相同。不同之处是,在搜索任务队列时不是参考每个任务的绝对截止时间值 RT_time[ID],而参考的是任务当前优先级的值 DP_priority[ID]。

TimerDP.fired() 事件函数完成定期增加队列中任务优先级(priority)的工作,LIP 算法流程如下页图 8 所示。主体思想是,设置一个时长为 T 的定时中断,每当定时中断到来后,即启动中断处理程序,在中断处理程序中,依次将就绪任务队列中任务的优先级增加某个常数增量(本文取 2),如果该优先级值已达最大,则不再予以增加。

在 DP_queue 队列的调度应用中,要根据不同的应用场合,选择合适的定时器中断时间间隔。如果间隔选的过大,低优先级任务依然在很长时间内不能得到调度;如果间隔选的过小,则所有就绪任务的优先级会快速上升至最大,那么该算法会退化为 FCFS 调度策略。

2.2.3 超时任务调度

该调度策略中用到 2 个定时器,一个是用来动态设置动态优先级队列中任务的优先级;另一个则

是判断任务是否超时,可以结束进入阻塞状态或死循环状态的任务,防止系统瘫痪。任务调度流程如图 9 所示。

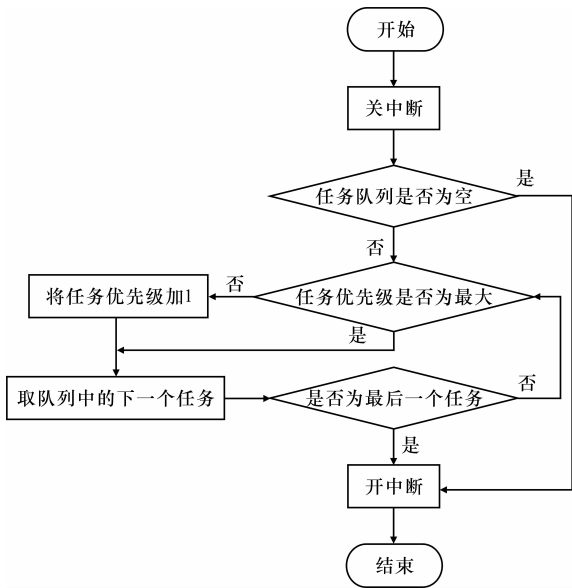


图 8 TimerDP.fired()中断调度流程

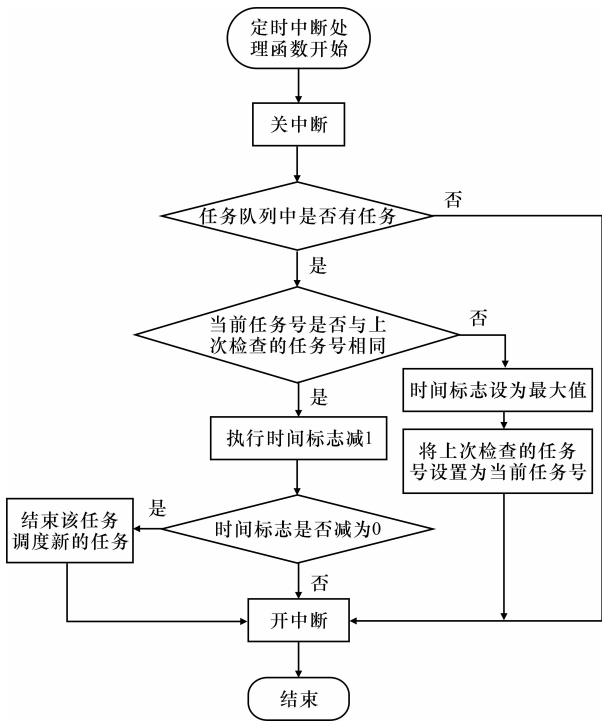


图 9 超时任务调度流程

3 系统测试与分析

以文中设计的无线传感器节点,搭建了车载环境下感知及数据发送测试环境,并进行了任务调度策略验证。

测试可以分为 3 个部分:一是实时任务调度分

析;二是优先级任务调度分析;三是调度策略的能耗分析。

3.1 实时任务调度分析

建立 ABC 3 个无线传感器节点,其中,节点 A 通过中间节点 B 向节点 C 发送数据包,A 的发送频率为 5 Hz;节点 B 除了负责转发数据包之外,还要运行本地任务。B 节点任务信息见表 1。

表 1 节点 B 的任务分配

类型	任务号	执行时间/s	周期/s
本地任务	Task1	30	150
	Task2	60	200
	Task3	90	250
	Task4	120	300
	Task5	150	300
转发任务	Task6	60	200
	Task7	80	200
	Task8	100	200
	Task9	120	200

假定任务 Task6~Task9 是时限值为 200 s 的实时任务,任务 Task1~Task5 为普通周期性任务,节点 B 首先运行 Task1,之后每间隔 1 s 就启动一个新的本地任务 TaskN,直至本地任务全部运行,通过节点 B 的数据包转发任务的完成时间来分析其实时性能。

图 10 为 TinyOS 自带的 FCFS 调度策略与改进后调度策略的试验结果对比。从图 10 中可以看出,改进后的调度策略可以及时响应实时任务,保证了任务完成的时限要求。

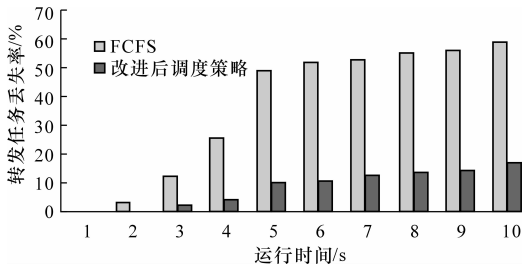


图 10 实时任务丢失率对比

3.2 优先级任务调度分析

建立 ABCDEF 6 个无线传感器节点,为节点 A 设计 5 个非实时任务 Task1~Task5,任务的初始优先级及周期见下页表 2。初始时任务发生频率均为 5 Hz,每个任务对应一个指定的节点发送数据包,在 6 s 时将任务 Task1~Task3 的发生频率改为 10 Hz,其他任务不变。通过统计接收节点收到的数据包数,分析不同调度策略下任务的执行情况。优先级任务调度策略如下页图 11~图 13 所示。

表 2 节点 A 的任务分配

任务号	初始优先级	执行时间/s	周期/s	目标节点
Task1	240	40	200	B
Task2	200	75	200	C
Task3	160	80	200	D
Task4	120	90	200	E
Task5	80	100	200	F

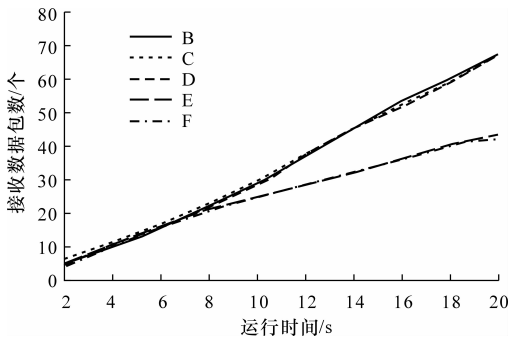


图 11 FCFS 任务调度策略

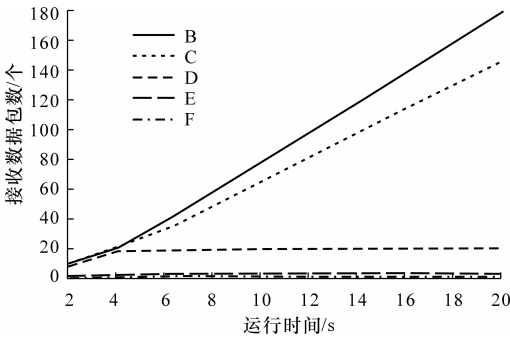


图 12 RM 静态优先级调度策略

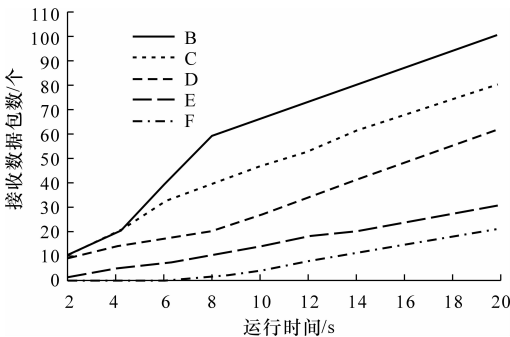


图 13 改进后动态优先级调度策略

从图 11 可知,FCFS 调度策略因为不考虑任务的优先级,因此对于那些重要任务并不能优先得到执行,影响系统对重要任务的响应速度。

由图 12 可知,静态优先级 RM 调度策略可以使得高优先级任务得到很好的调度,但如果碰到系统频繁提交高优先级任务时,会牺牲低优先级的调度机会,影响低优先级任务的吞吐量(表 3 对任务的吞吐量进行了计算)。

由图 13 可知,改进后的调度策略可以做到一个

很好的折中,兼顾了任务到达时刻和关键性任务优先调度,重要的任务可以被优先执行,同时低优先级任务在适当的时间之内也可以得到调度机会。

表 3 执行 20 s 后任务吞吐量对比

任务号	总投递任务数/个	RM 算法执行任务数/个	改进算法执行任务数	RM 算法吞吐量/%	改进算法吞吐量/%
Task1	180	179	101	99.4	56.1
Task2	180	146	81	81.1	45.0
Task3	180	20	62	11.1	34.4
Task4	100	3	31	3.0	31.0
Task5	100	0	21	0	21.0

3.3 调度策略能耗分析

对所设计调度策略能耗的评估,运用加州大学提供的在 PC 机上运行的 TOSSIM 仿真器进行仿真试验。TOSSIM 支持基于 TinyOS 应用程序模拟运行,在 TOSSIM 的基础上加入能量度量模块做成 Power-TOSSIM 模拟器,用于观察传感器网络运行的能耗情况。

设置 100 m×100 m 的仿真平面区域,在该区域内随机散布 N 个节点,使所有节点在规定的时间内完成本地数据采集任务和多跳路由转发任务等。在不同调度策略下,不断改变节点总数 N 值,利用每个节点的平均功耗来分析调度策略的能耗。相比 TinyOS 自带的 FCFS 调度策略,改进后的调度策略由于复杂度的增加,带来了一定的能耗损失,但是增加的能耗不大于 3%,远小于其他复杂高级的调度策略,如图 14 所示。

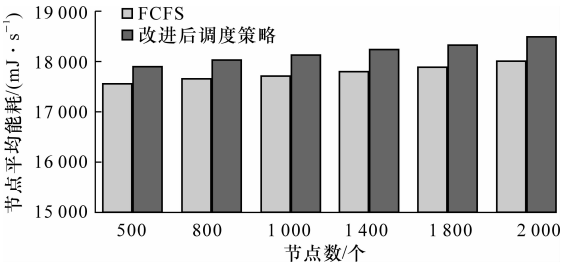


图 14 调度策略能耗比较

4 结 语

(1)任务调度器对非实时任务采取公平的动态优先级调度,可以确保低优先级任务也能在一定时间段后得到调度机会,提高了系统的吞吐量,而且加入的超时判断可以防止系统进入死循环。

(2)在任务固定、环境固定且能耗受限的情况下,采用固定路由及数据预判的方式可以减少通信冲突,降低整个网络的能量消耗,但存在关键节点损

坏会影响整个网络的传输,需要增加冗余中继节点来保证固定路由的可靠性,在一定程度上会使复杂度和成本增加。

参考文献:

References:

[1] Akyildiz I F, Su W, Sankarasubramaniam Y, et al. Wireless sensor network; a survey [J]. Computer Networks,2002,38(4):393-422.

[2] Mozumdar M M R, Luciano L, Vanzapo L. A comparison of software platforms for wireless sensor networks;MANTIS, TinyOS, and ZigBee [J]. Transactions on Embedded Computing Systems,2009,8(2):123-129.

[3] Karlof C, Waqner D. Secure routing in wireless sensor networks; attacks and countermeasures [J]. Ad Hoc Networks,2003,1(2/3):293-315.

[4] Ager J, Clare L. An integrated architecture for cooperative sensing networks [J]. Computer Magazine, 2000,33(5):106-108.

[5] 林 恺,赵 海,尹震宇,等. 一种基于 TinyOS 的自适应双环调度策略[J]. 东北大学学报:自然科学版, 2007,28(7):985-988.

LIN Kai, ZHAO Hai, YIN Zhen-yu, et al. An adaptive double ring scheduling strategy based on TinyOS

[J]. Journal of Northeastern University; Natural Science,2007,28(7):985-988.

[6] Cormac D, Utz R, John H, et al. Adding preemption to TinyOS[C]//IEEE. Proceedings of the 4th Workshop on Embedded Networked Sensors (EmNets 2007). New York: IEEE Press,2007:88-92.

[7] Yan Zhao, Qianping Wang, Wei Wang, et al. Research on the priority-based soft real-time task scheduling in TinyOS[C]//IEEE. Proceedings-2009 International Conference on Information Technology and Computer Science (ITCS 2009). Kiev: IEEE Press, 2009: 562-565.

[8] 尹震宇,赵 海,徐久强,等. 无线传感器网络操作系统中抢占式任务调度策略[J]. 东北大学学报:自然科学版,2007,28(5):652-655.

YIN Zhen-yu, ZHAO Hai, XU Jiu-qiang, et al. Pre-emptive scheduling strategy for wireless sensor networks operating system[J]. Journal of Northeastern University; Natural Science,2007,28(5):652-655.

[9] Zhibin Zhao, Zhuoyue Li, Shurong Liu, et al. Study on preemptive real-time scheduling strategy for wireless sensor networks[C]//IEEE. Proceedings of the 2008 International Conference on Advanced Infocomm Technology (ICAIT'08). New York: IEEE Press, 2008:101-106.

.....

(上接第 80 页)

参考文献:

References:

[1] 中国人民共和国交通管理局. 中华人民共和国道路交通事故统计年报 2008 年度[R]. 北京:公安部交通管理局,2009.

[2] 裴玉龙,程国柱. 高速公路车速离散性与交通事故的关系及车速管理研究[J]. 中国公路学报,2004,17(1):74-78.

PEI Yu-long, CHENG Guo-zhu. Research on the relationship between discrete character of speed and traffic accident and speed management of freeway [J]. China Journal of Highway and Transport, 2004, 17(1):74-78.

[3] Neuman T R, Ronald P, Slak K L. Guidance for implementation for the AASHTO strategic highway safety plan; a guide for reducing collisions on horizontal curves[R]. Washington DC: Transportation Research Board,2004.

[4] MUTCD-FHWA, Manual on uniform traffic control devices[S].

[5] Chowdhury M A, Warren D L, Bissell H. Analysis of advisory speed setting criteria[J]. Public Roads,1991,

55(3):23-26.

[6] 周蔚吾. 道路交通标志标线设置技术手册[M]. 北京:水利水电出版社,2007.

[7] JTJG D82—2009,公路交通标志和标线设置规范[S].

[8] 孙小端,陈永胜,贺玉龙. 从京津塘高速公路安全评价看中国高速公路交通事故特点及安全改进措施[C]//交通部公路科学研究院. 国际公路安全研讨会论文集. 北京:人民交通出版社,2005:32-36.

[9] Bonneson J, Pratt M, Miles J, et al. Development of guidelines for establishing effective curve advisory speeds[R]. Texas: Texas Transportation Institute,2007.

[10] 杜志刚,崔显忠. 山区公路典型小半径曲线安全改造[J]. 浙江交通职业技术学院学报,2008,9(3):4-8.

DU Zhi-gang, CUI Xian-zhong. Safety reconstruction of typical small-radius curve in mountain highways [J]. Journal of Zhejiang Institute of Commucnication, 2008,9(3):4-8.

[11] Venglar S, Porter R, Ballard A. Establishing advisory speeds on freeway exit ramps[C]//TKB. TRB Annual Meeting. Washington DC: Transportation Research Board,2010:55-60.

[12] 王建军,钟厚冰,赵晓峰. 道路交通标志设计理论与方法[M]. 北京:科学出版社,2008.