基于TinyOS的HRRF调度策略

李海成

(辽东学院信息技术分院 丹东 118003)

摘 要 针对传感器网络操作系统TinyOS采用非剥夺的先来先服务调度策略,系统紧急任务不能得到及时响应及节点吞吐量下降的情况,提出了一种可抢占HRRF(Highest-Response-Ratio First)作业调度策略。HRRF算法采用对于实时性较强的任务优先调度策略,满足了系统对实时任务的响应,提高了处理器的响应速度;对于软实时任务采用高响应比(任务等待时间/需运行时间)调度策略,提高了系统的效率。在TinyOS上的测试表明,HRRF策略在不影响TinyOS原有性能的情况下极大改善了传感器网络承担实时性任务的运行效果。

关键词 TinyOS操作系统,HRRF调度策略,可抢占,余实时性

中图法分类号 TP393 **文献标识码** A

Research on HRRF Scheduling Strategy Based on TinyOS

LI Hai-cheng

(Information Technology College, Eastern Liaoning University, Dandong 118003, China)

Abstract Currently the TinyOS task scheduling is based on first-come-first-served non-preempting strategy, which is not able to give emergency tasks quick response, and throughput of nodes is lower. To address this issue, this paper proposed a preempting algorithm, HRRF (Highest-Response-Ratio First) Algorithm. Through executing real-time tasks within an higher priority, it met system requirement for the real-time response, and improved processor's response speed. Through executing soft real-time tasks based on Highest-Response-Ratio (waited time/run time) scheduling priority strategy, system throughput was increased. The TinyOS test result indicated that under the condition of not affecting (decreasing) the original performance, HRRF strategy greatly improved sensor network performance in working on real-time tasks.

Keywords TinyOS, HRRF algorithm, Preemptive scheduling, Real-time

TinyOS[1]是美国加州大学伯克利分校专为网络嵌入式系统定制的操作系统。TinyOS采用基于事件驱动、两层调度的并发模型[2,3]。为了实现对多样化的无线网络传感器应用的支持,TinyOS是基于构件方式组成的,主要由主控构件(包括调度器)、应用构件、系统构件和硬件抽象构件组成。TinyOS的系统框架如图1所示。

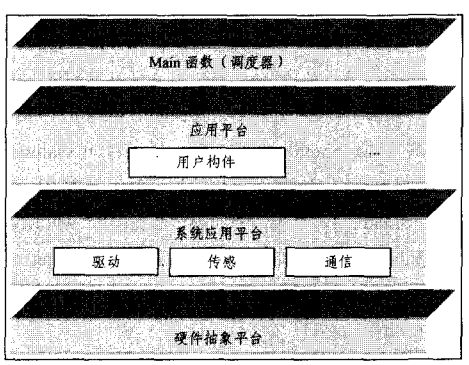

图1 TinyOS系统框架图

TinyOS在任务调度上采用非剥夺的先来先服务(first-come-first-served, FCFS)调度策略,不能对实时性很强的网络任务做出及时的响应[4]。针对此问题,本文提出一种结合实时性和高响应比(等待时间/需运行时间)要求的HRRF调度策略,并在TinyOS上加以实现。仿真试验表明,HRRF调度策略在不影响TinyOS原有性能的情况下,可以满足任务完成的实时性要求。

1 TinyOS调度机制分析

TinyOS提供任务加事件的两级调度,事件在中断服务例程中执行。同时TinyOS把一些不需要在中断服务程序中立即执行的代码以函数的形式封装成任务,任务一般用于对时间要求不高的应用中,它实际上是一种延迟计算机制[5,6]。任务由事件产生,在中断服务例程中将任务函数地址放入到任务队列,在中断服务例程退出后调度执行。任务之间互相平等,没有优先级之分,所以任务的调度采用简单的FIFO。任务间互不抢占,而事件(大多数情况下是中断)可抢占。即任务一旦运行,就必须执行至结束,当任务主动放弃CPU使用权时才能运行下一个任务[7],所以TinyOS实际上是一种不可剥夺型内核。内核主要负责管理各个任务,并决定何时执行哪个任务。

到稿日期:2009-06-04 返修日期:2009-08-13 本文受国家自然科学基金(69873007),辽宁省教育厅科学基金项目(2008212),辽宁学院科学基金项目(2009-Z03)资助。

李海成(1966—),男,副教授,主要研究方向为嵌入式系统和自动控制等,E-mail:lhc6607@126.com。

TinyOS 调度模型有以下特点[8,9]：

(1)任务单线程运行到结束，只分配单个任务栈。

(2)没有进程管理的概念，对任务按简单的 FIFO 队列进行调度。

(3)为了确保底层任务的执行不被长时间地延迟，个体任务的执行时间必需很短，耗时很长的操作必需分布在若干个任务中完成，增加了复杂度。

(4)任务之间互相平等，没有优先级的概念。

(5)某些任务(例如安全应用的加解密任务)执行时间很长，这时如果某些实时任务在该任务之后才进入任务队列，就会影响实时性；对于数据包的收发，就会影响波特率。

(6)任务队列中某个任务如果意外出现阻塞或异常，就会影响后续任务的运行，甚至导致系统瘫痪。

尽管 TinyOS 被广泛使用，并且得到了相当的认可，但这并不意味着 TinyOS 能够适用于无线传感器网络的所有应用场景。事实上，在某些场合下，TinyOS 并非工作得很好，而可能出现过载，导致任务丢失、通信吞吐量下降等[10]。

2 HRRF 调度策略实现

为了改善系统对紧急任务的响应，首先根据任务重要性的不同，HRRF 算法把任务分为硬实时任务和软实时任务两种。硬实时任务具有硬时间约束，其截止期的错过将会导致较严重的后果。软实时任务则具有软时间约束，其截止期的错过在某种程度下是可以被接受的，不会引起非常严重的后果。

2.1 定义任务数据结构

任务的数据结构是操作系统中使用频率最高的数据结构，它的作用就是内核通过此数据结构控制和管理此任务。新的定义如下：

```
typedef struct {
  void ( * tp)( );
  uint16_t deadline;
  uint16_t wtime;
  uint16_t runtime;
  uint16_t SP;
bool preempted;
} TOSH_sched_entry_T;
```

tp 字段是任务函数入口地址，任务的运行就是通过运行该入口地址的函数来完成的。

deadline 字段是新增加的字段，该字段不为空，表示该任务是硬实时任务，具有时限要求；该字段为空，表示该任务是软实时任务，没有时限要求。相对于当前，距离用户要求的任务的绝对时限还有时钟滴答数，是个相对的时限。

runtime 字段是新增加的字段，表示用户估计的任务的运行时间，单位也是时钟滴答。

wtime 字段是任务等待时间，初始值为 0，使用时钟滴答数计数。

SP 字段是本文加上的字段，表示任务当前的堆栈指针位置，主要用于发生中断时恢复现场。

preempted 字段是新增加的字段，表示当前任务是否被中断，若为 1 则表示该任务被中断，再次运行前需要恢复现场操作。

2.2 任务提交实现

TOS_post()函数的作用就是将任务提交到任务队列中，如果成功放入队列就返回 TRUE，否则返回 FALSE。TOS_edf_post()函数的算法如下：判断新任务的 deadline 值，若不为空，按照 deadline 值由小到大插入到队列行头部相应位置，如果 deadline 值为空，按照响应比(β=wtime/runtime)值由大到小插入到队列尾部相应位置上(即高响应比优先)，队列最后排序如图 2 所示。所有时限 deadline 值不为空的任务排列在所有 deadline 值为空的任务前面；所有 deadline 值不为空的任务按照 deadline 值由大到小排序；所有 deadline 值为空的任务按照响应比 β 的值由大到小排列；当有新任务提交时，首先判断 deadline 是否为空，然后任务队列按照 deadline 或者响应比值插入到任务队列相应位置上。

…	D(5)	D(8)	D(10)	D( )	D( )	D( )
	β(5)	β(4)	β(3)	β(2)	β(1)	β(0.5)

图 2 HRRF 任务队列

2.3 HRRF 调度策略

每次调度定时器中断时，更新任务的响应比值，选择任务队列头的任务，比较它与当前任务的响应比即比较二者的优先级，进行 HRRF 可剥夺式调度，当任务队列头的响应比高于当前任务的响应比，即任务队列头的任务优先级高于当前任务的优先级时，进行抢占，保持现场，切换任务运行的堆栈，改变被中断的任务的中断状态位，判断队列头高优先级任务的中断状态位，如果未被中断过，直接运行这个任务的处理函数。若该任务被中断过，需要先对这个任务恢复现场，然后运行任务的处理函数，直到下一次中断的到达。

修改后 HRRF 调度算法。提交任务时，同时提交任务的其它属性，如运行时间和任务时限等。任务的优先级根据其它任务时限和等待时间/运行时间比值来确定。每个新任务到达时，在任务队列中按照优先级排序，优先级高的任务排在任务队列头，处理器处理完当前任务后从任务队列头取出最高级的任务运行，每个任务动态更新任务时限，动态调整任务优先级。

3 性能仿真

对 HRRF 算法的评估，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，并在 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。仿真区域设定为 100m×100m 的平面区域。首先通过应用程序 SenseTask 测试采用 HRRF 和 FCFS 算法时系统的吞吐量百分比和能量消耗百分比，测试数据分别如图 3 和图 4 所示。

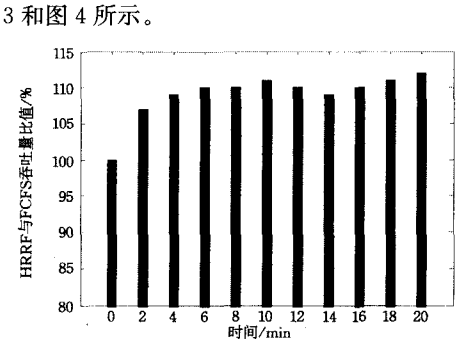

图 3 HRRF 和 FCFS 吞吐量百分比

(下转第 98 页)

分簇算法相比，MSEBNECHSA 能够以较低的能量成本将网络中的节点有效分簇，并且平衡网络中的能量消耗。仿真结果证明了该算法的这些特点。

参 考 文 献

[1] Heinzelman W, Chandrakasan A, Balakrishnan H. Energy- efficient communication protocol for wireless micro-sensor networks [C]// Proceedings of the 33rd Hawaii International Conference on System Sciences. 2000: 3005-3014

[2] Heinzelman W, et al. An application - specific protocol architecture for wireless microsensor networks [J]. IEEE Trans on Wireless Communications, 2002, 1(4): 660-670

[3] Younis O, Fahmy S. Heed: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks [J]. IEEE Trans. on Mobile Computing, 2004, 3(4): 660-669

[4] Ye M, et al. EECS: An energy efficient clustering scheme in wireless sensor networks[C]// Proc. of the IEEE Int'l Performance Computing and Communications Conf. New York: IEEE Press, 2005: 535-540

[5] 刘林峰，刘业. 一种无线传感器网络拓扑的启发式分簇控制算法[J]. 计算机研究与发展，2008，45(7)：1099-1105

[6] 胡静，沈连丰，宋铁成，等. 新的无线传感器网络分簇算法[J]. 通信学报，2008，29(7)：20-26

[7] 李建波，黄刘生，徐宏力，等. 一种密集部署传感器网络的分簇算法[J]. 计算机研究与发展，2008，45(7)：1106-1114

[8] Jindal A, Psounis K. Modeling Spatially Correlated Data in Sensor Networks [J]. ACM Transactions on Sensor Networks, 2006, 2(4): 466-499

(上接第 81 页)

从图 3 中可以看出，采用 HRRF 算法，系统总的吞吐量提高了 10%左右，因为对于软实时任务采用高响应比优先调度策略，单位时间内提高了系统的吞吐量。

图 4 显示，随着时间推移，系统总的能量消耗大约提高了 3%左右，虽然任务调度要消耗系统能源，但是调度耗能只占系统一小部分，传感器网络能耗主要消耗在通信模块中，HRRF 调度策略所带来的额外开销是很小的，适合应用在能源受限的无线传感器网络中。

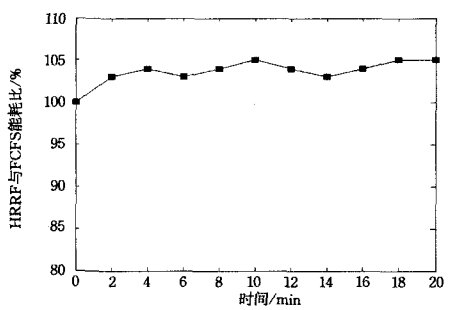

图 4 HRRF 和 FCFS 能耗百分比

图 5 显示了 10 个传感器节点在 1000s 内运行 Sense 时成功执行任务的比率。可以看出，HRRF 调度策略在面临强实时性任务时，任务成功执行的比率远高于 TinyOS 原有的 FCFS 策略。

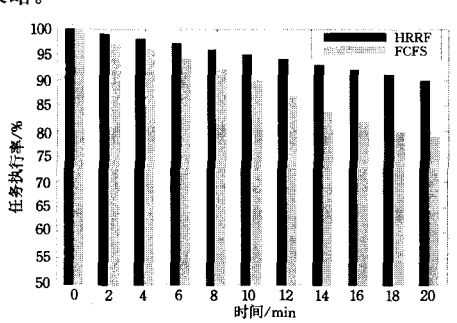

图 5 HRRF 和 FCFS 任务执行成功率和时间对比

结束语 TinyOS 作为一种典型的传感器网络操作系统被广泛使用，但是其 FCFS 调度策略使得 TinyOS 在强实时性及系统吞吐量方面受到限制。本文提出的 HRRF 调度策略，优先调度具有时限要求的任务，提高了系统对紧急任务的及时响应，同时对没有时限要求的任务综合考虑了其要运行时间和已经等待时间，采用高响应比调度策略，提高了系统吞吐量，改善了通信状况，保证了节点的高实时性和高吞吐量。仿真结果证明，该算法适合应用在能源受限的无线传感器网络中。

参 考 文 献

[1] Hong W, Madden S. TinySchema: creating attributes and commands in TinyOS[EB/OL]. [2005-03-11]. http: // webs. cs. berkeley. edu/tos

[2] Rmer K, Mattern F. The design space of wireless sensor networks[J]. IEEE Wireless Communications, 2004, 11(6): 54-61

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

[4] Vuran M C, Akan O B, Akyildiz I F. Spatio-temporal correlation: theory and applications for wireless sensor networks[J]. Computer Networks, 2004, 45 (3): 245-259

[5] Madden S, Franklin M J, Hellerstein J M, et al. TinyDB: an acquisitional query processing system for sensor networks [J]. ACM Transactions on Database Systems, 2005, 30(1): 122-173

[6] Intanagonwiwat C, Govindan R, Estrin D. Directed Diffusion: a scalable and robust communication paradigm for sensor networks[J]. Proceedings of the ACM MobiCom'00, 2000(1): 56-67

[7] Venkita S, Huang Huangming, Seema D, et al. Priority scheduling in TinyOS: a case study[R]. TR: WUCSE-2003. Washington University, 2003: 74

[8] Madden S, Franklin M J, Hellerstein J M, et al. TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks[M]. OSDI, Boston, USA, Dec. 2002

[9] Kuo S P, Kuo H J, Tseng Y C, et al. Detecting movement of beacon in location-t racking wireless sensor networks[A]// Proceedings of 2007 IEEE 66th Vehicular Technology Conference [C]. Baltimore, USA, 2007: 362-366

[10] Sivrikaya F, Yener B. Time synchronization in sensor networks: a survey[J]. IEEE Network, 2004, 18(4): 45-50