

嵌入式对称多核操作系统的设计与实现

赵立伟¹, 贺红卫², 刘冰²

(1. 中国航天科工集团第二研究院 706 所, 北京 100854;

2. 中国兵器工业计算机应用技术研究所, 北京 100089)

摘 要: 为解决嵌入式设备对嵌入式对称多核操作系统的需求, 设计基于 TMS320C6678 的对称多核操作系统。介绍多核操作系统的起源以及国内外的研究现状, 分析多核操作系统研究中所面临的一些主要问题, 总结多核操作系统的一些研究方法; 依据 TMS320C6678 处理器的硬件特性, 详细介绍该操作系统。结合实例验证了该系统的可行性和高效性。

关键词: 嵌入式; 多核操作系统; 任务调度; 内核同步; 对称多核

中图分类号: TP316.2 **文献标识码:** A **文章编号:** 1000-7024 (2014) 08-2741-05

Design and implementation of embedded symmetric multiprocessor operating system

ZHAO Li-wei¹, HE Hong-wei², LIU Bing²

(1. Institute 706, Second Academy of China Aerospace Science and Industry Corporation, Beijing 100854, China;

2. North Institute of Computer Application, Beijing 100089, China)

Abstract: To meet the need of embedded equipments for embedded symmetric multi-core operating system, a design of polynuclear operating system based on TMS320C6678 was presented. The development of embedded operating system was reviewed, and the origin of nuclear operating system and the present research status in domestic and foreign were introduced. The major problems in polynuclear operating system were analyzed and some research methods were summarized. Finally, based on the features of TMS320C6678 processor hardware, the proposed polynuclear operating system was described in detail. Combining with examples, the feasibility and effectiveness of the operating system were proved.

Key words: embedded system; polynuclear operating system; task scheduling; kernel synchronous; SMP

0 引 言

早先的操作系统和软件对于多核处理器的支持并不完美, 不能充分地利用多核处理器资源, 所以如何开发高效率、高稳定性多核操作系统是如今计算机操作系统发展的重要方向, 而支持 SMP 的嵌入式多核操作系统由于其共享内存, 负载平衡、功耗比高等优点, 更是成为嵌入式操作系统发展的重中之重, 因此对于 SMP 多核操作系统的研究具有十分重要的意义。从目前来看, 多核操作系统的发展严重滞后于多核处理器的发展, 同时多核操作系统性能的落后反过来也严重制约着计算机整体性能的大幅提升。国外操作系统的发展相对国内要早很多, 早在 1974 年, 卡内基梅隆大学便提出了 HYDRA 操作系统, 自此以后国外的

各种操作系统如雨后春笋般出现, 各大高校以及研究机构开始投入大量人力财力进行操作系统和多核处理器的研究。相比之下, 而国内操作系统的研究则起步比较晚, 能够进行深入研究的机构相对少很多, 虽然也取得了一些成就, 但是相对于国外还是落后很多。目前在国内外公开文件上, 面向高性能 DSP 多核处理器的操作系统还没有出现, 很多国家和公司也已经投入大量精力开始着手研究, 所以基于 DSP 的嵌入式对称多核操作系统必将是今后发展的一个重要趋势。因此在核高基重大课题专项的支持下, 开始进行基于 DSP 多核处理器的嵌入式对称多核操作系统的研究。本文对多核操作系统的研究方法进行了总结, 并在前人研究成果的基础上进行了扩展与补充, 使之能够满足 DSP 对于处理数据高效性的要求, 由此设计出了一种基于

收稿日期: 2013-09-09; 修订日期: 2013-11-15

作者简介: 赵立伟 (1987-), 男, 河北沧州人, 硕士, 研究方向为嵌入式操作系统; 贺红卫 (1970-), 男, 山西孝义人, 研究员, 研究方向为软件测试、软件开发环境; 刘冰 (1983-), 男, 山东聊城人, 硕士, 研究方向为嵌入式操作系统、软件开发环境。

E-mail: zlw5307@163.com

TMS320C6678 的对称多核操作系统。

1 多核操作系统的关键问题

相对于单核操作系统,多核操作系统的设计存在如下几个主要难点:一是多核处理器的运行模式。二是临界资源的访问控制,必须采取某种策略实现对多核处理器临界区资源的互斥访问。三是任务调度,必须设计出性能良好的调度算法以满足操作系统的实时性要求、同时还要保证处于最高优先级的任务处于运行状态。四是多核 cache 一致性的管理。同时多核操作系统也应该包括单核操作系统的所有功能与模块,下文主要对各个技术难点进行详细介绍。

1.1 多核操作系统的运行模式

多核操作系统的运行模式包括以下 2 种:

非对称多处理 (asymmetric multiprocessing, AMP): 在每一个 CPU 内核独自运行着一个操作系统,各个核上的操作系统互不干涉,通过任务通信算法或者核间中断来实现各个核之间的数据通信。

对称多处理 (symmetric multiprocessing, SMP): 一个操作系统内核负责管理所有的处理器核。同时所有的任务可以运行在每一个 CPU 核上。

在 AMP 系统中,一个任务 (process) 只能运行在某个特定的 CPU 核上面,即便是其它内核正在处于空闲状态。这样会导致某些内核处于空闲状态,而另外一些核在超负载运转,即负载不平衡。所以 AMP 系统虽然也能使多核操作系统正常的运转,但却不能最大程度地利用多核处理器的资源,不能充分发挥其性能。然而一个设计良好的 SMP 操作系统可以允许多个任务协同地运行在任何一个内核上,这种协同性可以充分地利用多核处理器资源,使操作系统的性能大幅提高。所以 SMP 操作系统必定是多核操作系统发展的重要方向。

1.2 临界资源的访问控制

多核处理器是一个多任务系统,其主要特点是资源的共享,为使操作系统能够正确运行,需要维护一些共享资源的互斥访问机制。而传统的用于单核的共享资源的互斥访问都是通过关中断等方式实现,显然这种解决机制并不能满足多核 CPU,因此需要利用硬件提供的“读-修改-写”的原子操作或其它同步互斥机制来实现。

1.3 任务调度

任务调度的方式分为剥夺方式和非剥夺方式 2 种。非剥夺方式:一旦某个任务获得了 CPU 资源后就会一直运行下去,直到任务完成或发生任务调度等事件时,其余的任务才会获得 CPU 资源。剥夺方式:操作系统可以基于某些原则,把一个正在运行任务的 CPU 资源剥夺,并将其分配给其它任务。剥夺原则有:优先权原则、短任务优先原则、时间片原则。

任务调度策略:把处理器资源分配给按照一定的规则

从就绪队列中选取的任务,使其开始运行。这些选取任务的规则就是任务调度策略,其基本要求是高效、公平。常见的任务调度策略有:优先级高优先调度策略、多重循环轮转调度策略、时间片轮转调度策略等。

任务队列模型分为全局队列模型和局部队列模型 2 种方式。局部队列模型是每个核维护自己的任务就绪队列,这样由于就绪队列是每个核所私有,不存在互斥访问的问题,所以多个核之间可以并行的进行任务调度,具有较高的吞吐量。但是却不能实现任务的负载平衡,需要操作系统维护额外的任务负载平衡算法,并且不能保证高优先级的任务一定能够处于运行状态,可预测性也比较低。全局队列是所有核拥有同一个任务就绪队列,这样在每个核必须互斥访问任务就绪队列,因此在同一个时刻只能有一个核进行任务调度,吞吐量较低。但是可以保证优先级最高的几个任务运行在这些 CPU 核上,与局部队列模型相比,可预测性有了很大提高,而且由于所有核共享同一个任务就绪队列,每个核无需维护额外的任务负载平衡算法。

1.4 cache 一致性管理

多核 cache 一致性是指,当某个核把部分数据加载到 cache 中运行时,如果运行过程中对某些全局变量进行了修改,而内存中和其它核的 cache 中的数据并没有随之而改变,会造成 cache 与内存数据的不一致性,该问题将会导致程序执行错误,甚至引起系统崩溃。因此需要对 cache 的一致性进行维护。同时,一个性能良好的 cache 一致性管理策略对于操作系统的性能具有相当大的影响。

目前实现 cache 一致性的策略主要分为硬件 cache 一致性策略和软件 cache 一致性策略。其中硬件 cache 一致性策略又分为监听 cache 一致性协议和目录表法等。

2 基于 TMS320C6678 的 SMP 多核操作系统设计

2.1 启动的设计

在多核处理器的引导过程中,处理器内核间是不平等的,有主次之分的。系统启动后,PM (primary CPU) 首先对其自身进行初始化,然后完成全局的初始化(如 PLL、内存控制器初始化等),向各个 PE (processor element) 发送核间消息,通知各个 PE 进行启动。从核在接收到消息之后便开始对其自身进行初始化,同时在 PE 初始化的过程中,PM 需要循环地检测是否所有处理器核都完成了初始化,如果已经全部完成,就可以同时进入任务调度过程。

在引导过程中,所有的处理器核都需要进行自身的初始化过程,而全局的初始化过程只需要 PM 执行一次,PE 无需执行。

SMP 启动流程如图 1 所示。

2.2 共享资源的互斥访问

我们没有方法控制对共享资源访问的有序性,但是有能力对共享资源采用锁的保护机制,当某个共享资源被锁

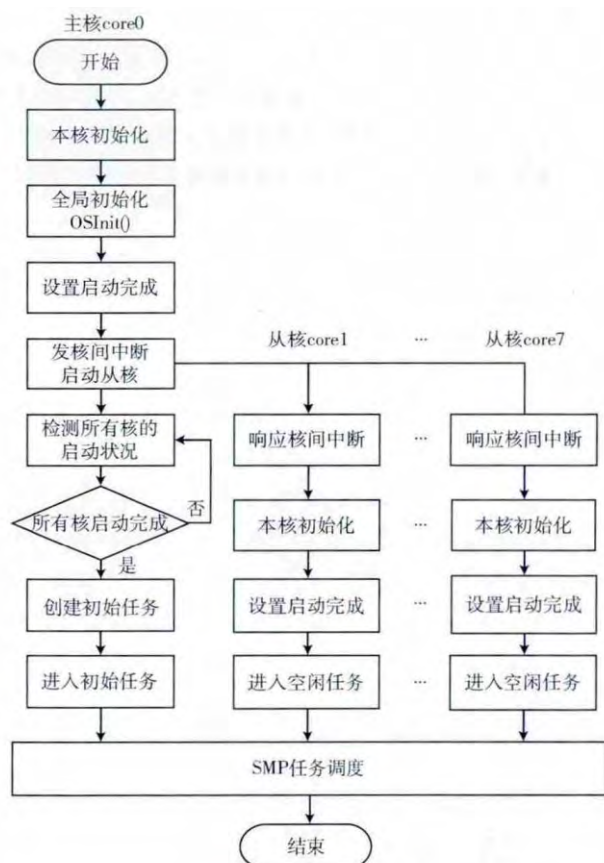


图 1 系统启动流程

住时, 只有获取该锁的 CPU 核能够操作共享资源, 其余试图访问共享资源的 CPU 核只能等待这个锁的释放, 这就是自旋锁的保护机制。

自旋锁的设计思想是基于 Test-and-Set 机制, 对此 C6678 则提供了一组 (32 个) 硬件信号量 semaphore, 通过对相应的寄存器的读写来保证互斥地获得该信号量。

C6678 硬件信号量包括 3 种工作方式, 分别为: direct 方式、indirect 方式和 combined 方式, 我们选择 direct 方式对自旋锁进行实现, 具体流程如图 2 所示。

2.3 负载均衡的任务分配与调度

多核嵌入式操作系统的负载均衡管理是指依据某种管理机制将所有任务在各个核上进行分配与调度, 实现各个核的任务负载均衡, 每个核的任务量大致相当, 不会出现某些核空闲, 而某些核超负荷运作的情况。负荷的有效管理能最大程度的发挥多核嵌入式处理器的并行计算资源, 最大程度的实现处理器的有效利用。

同时任务调度是操作系统最重要的一个模块, 因为当新任务创建、任务延时、请求/释放信号量、中断返回时, 不可避免的要采取任务调度, 而上述活动却又要频繁发生的, 因此, 选择一个好的任务调度策略对于操作系统的性能具有非常重要的意义。

对称多核操作系统的任务管理与调度分为 2 种模式,

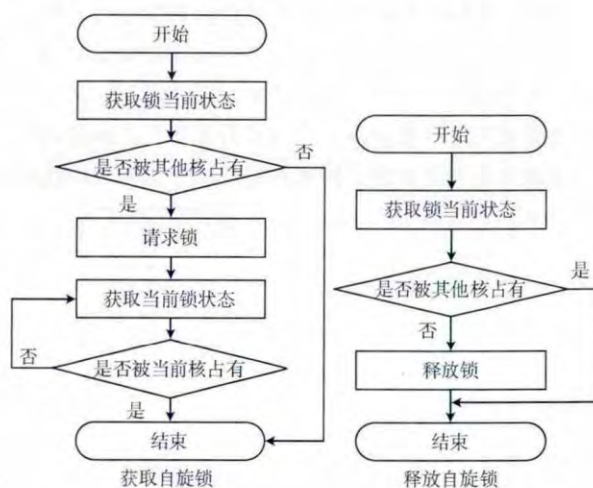


图 2 自旋锁获取与释放流程

即: 全局队列模式和局部队列模式, 全局队列模式是指所有的处理器核共同维护一个全局任务就绪队列, 当有一个或者若干个核空闲时, 操作系统按照某种管理机制从全局队列中取出一个或者若干个任务分配到相应核上运行, 完成任务的动态分配与调度管理。局部队列模式是为每个任务维护一个局部任务就绪等待队列, 当进行任务调度时, 每个核都需要从本核的任务就绪队列中选取任务。2 种模式的工作情况如图 3 和图 4 所示。

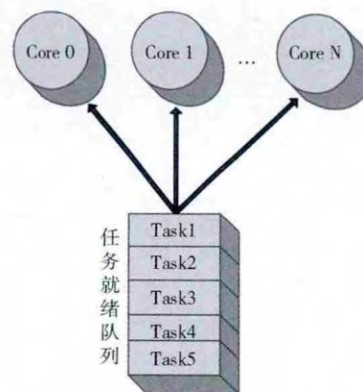


图 3 全局队列模式

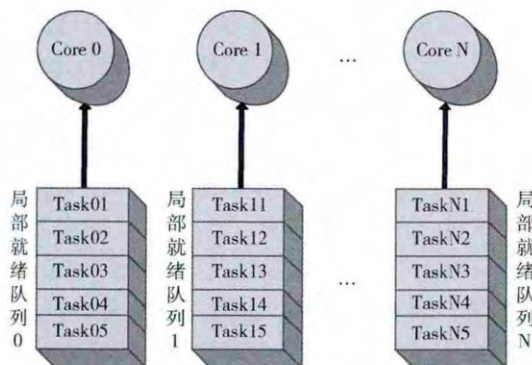


图 4 局部队列模式

显然,全局队列调度相对于局部队列调度具有较好的可预测性并且可以满足负载平衡,而这两者是实时系统中非常重要的指标,故本设计选择全局队列调度。同时为了满足最高优先级的任务永远处于运行状态,必须要选择可抢占式的任务调度策略,因此采用了剥夺方式中的优先级

高优先调度策略。

为了获得较高的执行效率,采用了2种实现方法,即:为某个核调度一个任务和为任务选择一个核。对此,任务延时、请求信号量、中断返回应该采用前者,而新任务创建、释放信号量则应该采用后者。2种实现方法的流程如图5所示。

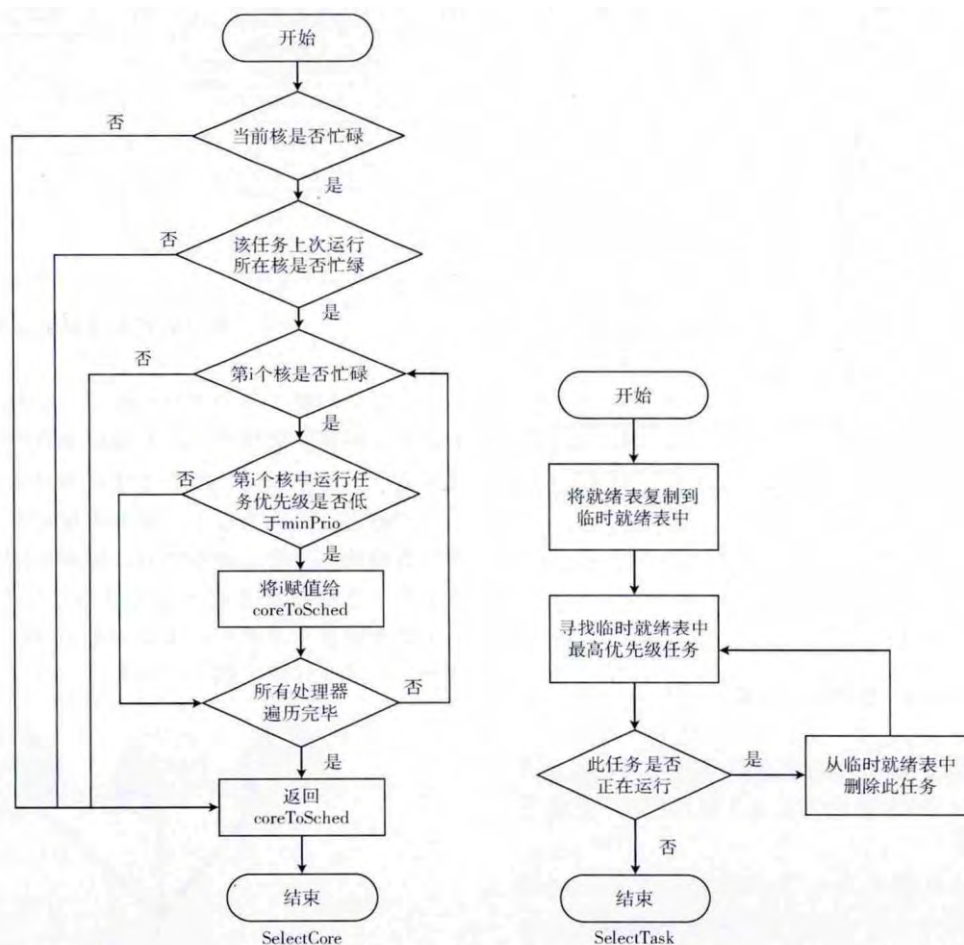


图5 任务调度流程

注:处于运行状态和就绪状态的任务都在任务就绪表中,此两者不同之处在于任务描述模块中有不同标示。

2.4 cache一致性策略的实现

目前大多数的多核处理器架构都提供硬件 cache 一致性维护策略,因此基于这种架构的多核操作系统就无需对其 cache 一致性进行软件维护。然而 TMS320C6678 并没有提供硬件 cache 一致性策略,因此需要软件来进行维护。

在讨论软件 cache 一致性策略之前,我们先对它的内存架构进行简单介绍。TMS320C6678 的每个核都有各自的一级和二级 cache,所有核共有共享内存空间。具体情况如图6所示。

软件解决的方案主要包括2种,一种方案是每个核的程序在执行过程中,如果需要访问共享变量,为了防止取到的是 cache 中的过时数据,则需要首先置对应 cache 无效,同时如果对全局变量进行了修改,则需要将相应的 cache 写回到共享内存中,以保证共享内存中的数据保持最

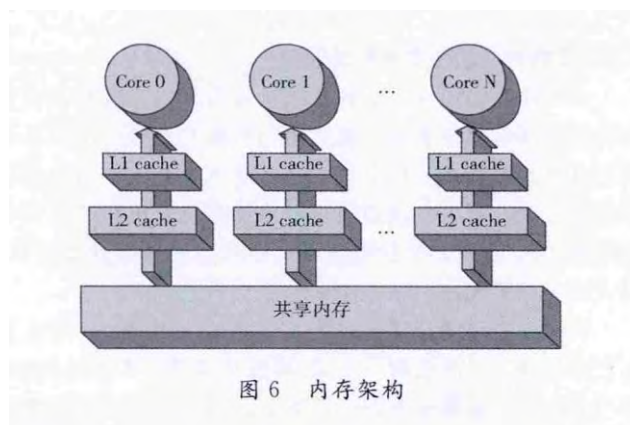


图6 内存架构

新状态。另一种方案是全局变量不加载到 cache 中,仅仅将其放在共享内存中,这样所有访问操作都要在共享内存中

进行。以上方案都只能采取较为保守的方法, 凡是可能发生不一致问题, 都将会使 cache 无效, 将会导致较高的 cache 不命中率, 影响着操作系统性能的进一步提升。但是相对于硬件维护策略, 它们也具有不随着处理器核的增多而变得复杂, 并且不需要很多辅助硬件等优点。

3 结束语

拥有了多核操作系统, 可以更大限度的发挥多核处理器的性能, 本研究实现的操作系统具有较强的实时性与健壮性, 结合 TMS320C6678 多核处理器的高速数据处理能力, 能够满足军用计算机对于多核操作系统的需求, 经过 TI 公司发布的 vlfft 性能验证程序验证, 本研究实现的系统与 TI 公司发布的 sys/bios 相比拥有更高的性能, 并且更加适合数据的并行处理与流水线处理, 可以更加完美的支持图像处理等应用程序。

参考文献:

- [1] Robert Love. Linux kernel development [M]. 3rd ed. Beijing: China Machine Press, 2010: 41-67.
- [2] Stallings W. Operating systems: Internals and design principles, 6/E [M]. Pearson Education India, 2009.
- [3] Boyd-Wickizer S, Chen H, Chen R, et al. Corey: An operating system for many cores [C] //OSDI, 2008: 43-57.
- [4] Yuan Q, Zhao J, Chen M, et al. GenerOS: An asymmetric operating system kernel for multi-core systems [C] //IEEE In-

ternational Symposium on Parallel & Distributed Processing, IEEE, 2010: 1-10.

- [5] Texas Instruments Inc. KeyStone semaphore2 hardware module [EB/OL]. <http://www.ti.com/cn/lit/ug/sprugs3a/sprugs3a.pdf>, 2011.
- [6] Texas Instruments Inc. Tms320c66x DSP CorePac user guide [EB/OL]. <http://www.ti.com/cn/lit/ug/sprugw0b/sprugw0b.pdf>, 2011.
- [7] Texas Instruments Inc. Multicore programming guide [EB/OL]. <http://www.ti.com/cn/lit/an/sprab27b/sprab27b.pdf>, 2011.
- [8] Kahn K C, Corwin W M, Dennis T D, et al. iMAX: A multiprocessor operating system for an object-based computer [C] //Proc of the 8th Symposium on Operating System Principles, 2009: 127-136.
- [9] Wulf W, Cohen E, Corwin W, et al. HYDRA: The kernel of a multiprocessor operating system [J]. Communications of ACM, 2009, 17 (6): 337-345.
- [10] Chunmao J, Guoyin Z, Chunmei H. Research of embedded operating system based on multi-core processor [C] //3rd IEEE International Conference on Computer Science and Information Technology, IEEE, 2010: 641-643.
- [11] Loghi M, Poncino M, Benini L. Cache coherence tradeoffs in shared-memory MPSoCs [J]. ACM Transactions on Embedded Computing Systems, 2006, 5 (2): 383-407.

(上接第 2708 页)

- [5] Himsoon T, Siri Wongpairat WP, Zhu Hane, et al. Lifetime maximization via cooperative nodes and relay deployment in wireless networks [J]. IEEE Journal on Selected Areas in Communications, 2007, 25 (2): 306-317.
- [6] BAI Zhiqun, JIA Jianlan, YANG Xianglai, et al. The large-grange algorithm based relay selection and power optimization in cooperative system [J]. Journal of Shandong University, 2012, 42 (3): 25-30 (in Chinese). [白智全, 贾建岚, 杨祥来, 等. 协作通信中基于 Lagrange 算法的中继选择和功率优化方案 [J]. 山东大学学报, 2012, 42 (3): 25-30.]
- [7] Chen Fuyu, Su W, Batalama S, et al. Joint power optimization for multi-source multi-destination relay networks [J] IEEE Transactions on Signal Processing, 2011, 59 (5): 2370-2381.
- [8] Rehman H-u, Wolf L A. Multihop IEEE 802.11 MAC protocol for wireless Ad Hoc networks [C] //29th IEEE Interna-

tional Conference Distributed Computing Systems Workshops, 2009: 432-439.

- [9] JING Zhengwei, XIE Yi, YUAN Pengfei. A cooperative MAC protocol with neighbor nodes information for Ad Hoc networks [C] //International Conference on Machine Learning and Cybernetics, 2012: 578-585.
- [10] LI Nan, CHEN Naiping, XU Xiaotie. ARC-MAC: An automatic relay cooperative MAC protocol for wireless Ad Hoc networks [C] //Wireless Communications & Signal Processing, 2009: 1-5.
- [11] ZHOU Yong, LIU Ju, ZHENG Lina, et al. Link-utility-based cooperative MAC protocol for wireless multi-hop networks [J]. IEEE Transactions on Wireless Communications, 2011, 10 (3): 995-1005.