

基于硬件多线程网络处理器功耗可控 无线局域网 MAC 协议实现^{*}

王 磊, 张晓彤, 张艳丽, 王 沁

(北京科技大学 计算机与通信工程学院, 北京 100083)

摘 要: 针对在如何在提高网络吞吐率并满足实时性需求的同时消耗更少的功耗的问题, 以硬件多线程网络处理为平台, 以 IEEE 802.11MAC 层协议为例, 通过对 MAC 层数据流的模式、数据流上的操作行为以及时间约束进行建模并测试分析, 提出一种多线程化网络协议的软件实现方法; 配合动态功耗可控的多线程网络处理器能够根据流量和实时性自适应地调整系统的性能。实验结果证明, 异构多线程结构程序在实时性任务时五个软件线程需四个硬件线程支持, 而无实时性任务只需两个硬件线程支持。提出的多线程 MAC 层协议编程模型能够达到根据网络负载特征动态控制处理器性能的目的。

关键词: 无线局域网; 编程模型; 实时性; 低功耗; 多线程

中图分类号: TP393.05

文献标志码: A

文章编号: 1001-3695(2012)07-2624-05

doi:10.3969/j.issn.1001-3695.2012.07.061

Power controllable WLAN MAC protocol implementation based on hardware multi-threaded network processor

WANG Lei, ZHANG Xiao-tong, ZHANG Yan-li, WANG Qin

(School of Computer & Communication Engineering, University of Science & Technology Beijing, Beijing 100083, China)

Abstract: How to improve network throughput and meet the real-time while consuming less power process are key concerns during network processor designing. Hardware multi-threaded network processor as a platform, IEEE 802.11MAC layer protocol as an example, modeling based on MAC layer data stream model, data stream operation behavior and time constraints and test the model. This paper presented a multi-threaded network protocol software implementation method. This method could adjust system performance based on traffic and real-time with dynamic power controlled multi-threaded network processor, thereby reducing power consumption when the processor was running. The result shows that real-time tasks require 4 hard threads on multi-threaded processor while only 2 are required for non-realtime tasks. This programming model provided processors the ability to dynamically adapt network workload characteristics.

Key words: WLAN; programming model; real-time task; low power; multi-threading

1 研究背景

飞速发展的网络应用促进了网络带宽的快速增长,同时也促进了大量的新协议与新服务的发展,从而对网络中用于发送接收数据并执行网络数据处理的传输节点的性能提出了更高的要求。而传统的网络设备一般采用 ASIC 或通用处理器,不能同时满足日新月异的网络发展所带来的对处理速度和灵活性的需求,为接入网的升级换代带来了巨大的负担。

网络处理器正是该背景下的产物,它兼顾了通用处理器的灵活性和 ASIC 的执行效率。目前的网络处理器内部一般都是多核结构,各个核模块以流水线方式组织在一起或者并行工作。同时,硬件多线程技术也被网络处理器引入以提高处理器的利用率。

网络快速发展所带来的对能量的消耗有了显著增加,功耗控制成为必须解决的问题。而如何在提高网络吞吐率并满足

实时性需求的同时消耗更少的功耗则是网络处理器设计过程中的另一个挑战;而网络系统中网络流量和实时性需求都是实时变化的,如何根据外部环境对系统设备需求的变化合理地调整系统以降低系统功耗则是一个研究热点。

网络处理器一般担负着网络体系结构中第二层到第七层数据报的处理任务。目前大多数研究侧重于对 IP 层及其以上层的研究。本文将对网络处理器中 MAC 层进行研究,网络处理器的 MAC 层负责从物理层接收网络数据包,处理后转给 LLC 层以完成上层网络处理,同时还负责转发 LLC 层传输过来的数据包进行相应处理以发送到外部物理层。网络处理器的高速处理能力、实时性和灵活性要求 MAC 层也必须具有高速的处理能力,满足网络 MAC 协议实时性需求同时兼具灵活性。

本文以 802.11MAC 层协议为例,介绍一种多线程化网络协议的软件实现方法,配合动态功耗可控的多核或多线程网络处理器能够根据流量和实时性自适应地调整系统的性能,从而

收稿日期: 2011-11-08; **修回日期:** 2011-12-09 **基金项目:** 国家“863”高技术研究发展计划资助项目(2008AA01Z134)

作者简介: 王磊(1984-),男,安徽合肥人,博士,主要研究方向为无线网络、嵌入式系统、体系结构(left.lei.wang@gmail.com);张晓彤(1968-),男,教授,博士,主要研究方向为无线传感器网络、体系结构;张艳丽(1985-),女,硕士,主要研究方向为无线网络、体系结构;王沁(1961-),女,教授,博士,主要研究方向为无线传感器网络、体系结构、集成电路。

降低处理器运行时的功耗。

2 国内外相关研究

网络处理器是一种专门面向网络协议处理的通信处理器。目前针对网络处理器的相关研究主要关注于网络处理器的体系结构,网络处理器性能分析模型,基于网络处理器的软件设计和应用等,对于网络处理器的低功耗研究较少。文献[1]在 SimWattch 模拟平台上,就频率动态调整和程序设计语言不同结构成分对应用程序运行功耗的影响进行了模拟和分析,通过对一组 Benchmark 程序的模拟,结果表明在编译系统、操作系统或应用程序设计中采用这些低功耗优化技术设计可降低至少 23% 以上的运行功耗。

目前一般采用软硬件协同设计方法使得处理器在处理实时复杂且多样的网络协议时满足性能需求同时降低功耗。文献[2]针对 IEEE 802.11a MAC 层网络协议提出一种软硬件协同设计方法,提出一种根据对系统响应速度需求、FPGA 门数量及未来升级灵活性等各方面因素进行软硬件实现功能的划分方法。文献[3]详述 PULSERS 项目中为低功耗、低复杂度和具转发功能的系统提出的基于 IEEE 802.15.4 的 MAC 层设计方法。文献[4]设计了一种适用于 IEEE 802.11a 标准 MAC 层和物理层的处理器芯片,MAC 协议完全由软件实现。对于处理器在 MAC 层的性能评估是通过其对物理层性能提高的贡献来评价的。文章中通过实现立即 PS-Poll 响应大大减少物理层能量消耗,并通过芯片的晶体管规模、工作频率、功耗、误差向量幅度以及接收信号强度调整时间等数据作为量化评价处理器整体性能的标准。文献[5]在 RPN 中可以在通过减少网络互连性的情况下减少系统功耗。文献[6]提出 MRRH 协议在保证无线网络低延迟的前提下降低功耗提高带宽。文献[7]提出根据网络流量特点动态调整设备电源管理状态以节省功耗。文献[8]通过流水线调节技术来减少功耗。文献[9]通过将大多数接收端任务转移到基带滤波器和放大器减少功耗。文献[10]提出时钟门限技术来减少高性能网络处理器的功耗。文献[11]提出了通过基于神经网络通信流量预测模型的互连网络通信链路电压调节算法来有效降低互连网络的功耗。

3 IEEE 802.11 MAC 协议及分析

3.1 协议内容分析

IEEE802.11 MAC 层主要提供了两种不同功能的媒体访问控制方式,即分布式协调功能(DCF)和点协调功能(PCF)。所谓媒体访问控制方式主要是决定什么时候哪个工作站发送数据的机制。DCF 是基本的媒体访问方式,主要利用载波监听多路接入和冲突避免(CSMA/CA)的技术来收发异步数据。PCF 是可选的媒体访问方式,在 DCF 基础上实现无竞争的访问。本文仅就 DCF 进行讨论。

IEEE 802.11 支持多种服务,主要分为针对数据业务的服务及与管理功能相关的服务。管理功能相关的服务主要包括认证、连接、取消认证、取消连接、重新连接等服务,数据业务主要提供站点间数据收发功能。MAC 层协议功能构成如图 1 所示。

图中每个模块利用下层模块提供的服务,为上层模块提供相应的服务。MAC 层数据业务模块、MAC 层管理业务模块、

MAC 层数据收/发模块构成了 MAC 层的核心部分,用来实现协议的基本功能。MAC 层核心部分还提供 MAC 层与无线网络 PHY 接口和 MAC 层与上层 LLC 层接口。本文只对 MAC 层数据收发进行重点讨论。

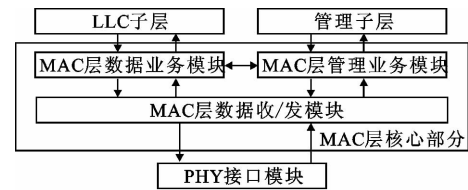


图1 MAC层协议系统框图

3.2 单线程协议实现方法

MAC 层数据收发单线程流程如图 2 所示。当 MAC 层收到 LLC 层发送过来的数据时,MAC 层接收来自 LLC 层的数据,封装成 MAC 帧格式 MSDU,并将 MSDU 根据目的地址和帧的长度进行分段,根据需要对数据加密形成 MPDU,如果采用 RTS/CTS,则需要先构建 RTS 并根据 DCF 媒体接入控制方式选择合适时机准备将帧发送到 PHY 层并为其生成 FCS;如果在 CTS 超时时间内收到 CTS 帧,则在退避过程后,为 MPDU 生成 FCS 域和时间戳,在收到 PHY 准备好的信号后将 MPDU 送往 PHY 层。

当 MAC 层收到 PHY 层发送过来的数据时,首先对其进行 FCS 重校验,随后进行帧过滤,主要检测帧的目的地址并避免重复帧;随后对帧头进行解析,根据帧类型分类处理,如果是数据帧,则发送 ACK 响应帧,并根据需要对数据解密和分段重组,最终将数据传给 LLC 层。

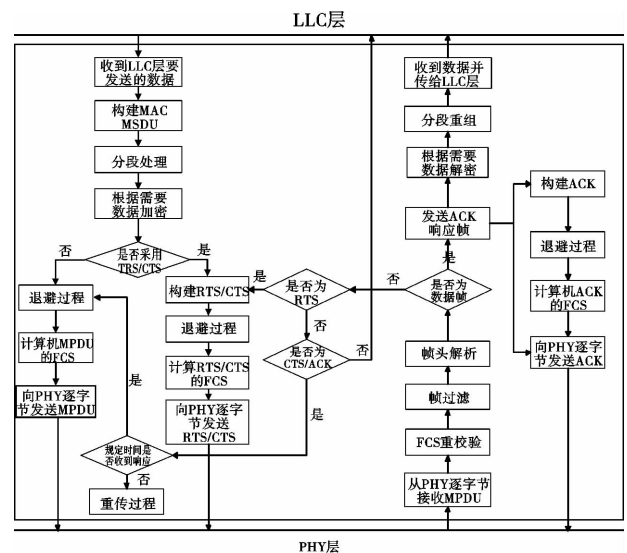


图2 无线站点STA的MAC层数据收发单线程流程

3.3 单线程协议实现的测试分析

本文利用兼容 MIPS32 指令集的模拟器构建 MAC 层协议处理器的行为级模型。用模拟器验证处理器功能的正确性,统计指令执行频度等。同时采用 C 语言在无操作系统环境下编写基本指令集支持的功能,用汇编方式编写和调用特殊指令集支持的库函数(如加速加解密的比特交换指令),以单线程结构来实现 802.11 MAC 协议数据处理的关键部分,经 MIPS 交叉编译器编译并在兼容 MIPS32 指令集的上述周期级精确微体系结构模拟器测试通过,该测试数据仿真参数如表 1 所示。最终测得数据如表 2 和 3 所示。

表1 协议仿真主要参数

参数	参数值	备注
Preamble	Long	前导码
tSlotTime	20 μ s	Slot 间隔时间
tSIFSTime	10 μ s	SIFS 间隔时间
tDIFSTime	50 μ s	DIFS 间隔时间
aCWmin	31	竞争窗口下限
aCWmax	1 024	竞争窗口上限
DataRate	54 Mbps	发送速率
Data-MPDU	2 346 Byte	MPDU 上限

表2 协议数据处理部分处理时间及指令数

指令条数	MAC-PHY/ PHY-MAC	其余发送处理过程 (无加解密)	其余接收处理过程 (无加解密)
RTS	6 374	10 259	7 299
CTS	6 246	8 845	7 306
DATA	55 155	79 933	33 522
ACK	6 250	8 240	6 866

表3 协议加密算法指令数

加密类型	加密指令条数	解密指令条数
WEP-RC4	194 862	213 002
CCMP-AES	1 165 326	1 148 302 ^[12]

通过对以上数据进行分析,在现有 MAC 层软硬件不变的情况下,单线程的实现方法存在以下问题:

a) 对网络吞吐量有所限制。当同时收到多个发送到该 STA 站点的帧,或者随着无线网络的发展,当无线网络速率越来越大时,MAC 层软件功能模块的处理速度较慢,由此可能导致数据丢失。由此可见 MAC 层软件功能模块将成为整个系统处理速度以及系统流量的瓶颈所在。

b) 无法满足网络实时性需求。当 MAC 帧长度越来越大时,为满足 MAC 协议的实时性需求(在规定时间内发出响应帧),网络处理器需要在短时间内处理几万条指令,这对网络处理器提出了巨大挑战。

4 基于硬件多线程的软件多线程 MAC 协议实现

为解决以上问题,本文提出一种基于硬件多线程的软件多线程 MAC 协议实现,以多线程实现任务并行,从而提高 MAC 层处理速度。网络处理器一般采用硬件多线程技术^[13],提高处理器的利用率。针对提高网络吞吐量和满足网络实时性两个需求,本文提出同构软件多线程和异构软件多线程的概念。

所谓同构软件多线程,即每个线程完成的功能都一样,但各个线程处理不同的数据。所谓异构软件多线程,即每个线程完成不同的功能,各个线程并行运行协同完成一项任务。

4.1 基于网络流量的同构多线程设计

对于 802.11 网络而言,影响网络流量的两个因素是网络速率和需要发送到目的站点的帧的总量,网络速率决定了一个帧占用信道的时间,网络速率越大,帧占用信道时间越短,单位时间内可能发送的帧越多,网络流量越大。网络中需发送到目的站点的帧的总量越多,帧与帧间的间隔时间越短,单位时间内收到的帧越多,网络流量越大。因此网络流量必定是动态实时变化的。当网络流量增大时,可能导致单线程 MAC 层软件处理不及时以致部分 MAC 帧不能被处理或者处理过程出错,从而降低了系统吞吐率。而当网络流量很少时,让整个网络处理器处于正常工作模式则会增加系统功耗。如何根据网络流量动态调整以使系统功耗最低并兼顾系统吞吐率是待解决的问题。

为此,本文提出以同构软件多线程来解决上述问题,如图 3 所示,当网络流量增加时,利用硬件多线程技术,采用同构软件多线程方法,同时开启多个线程,每个线程执行同样的任务;当接收到 PHY 层开始接收的信号时,开启一个线程对每个接收到的 MAC 帧进行相应的处理,尽量减少 MAC 帧占用系统缓存等待处理的时间,从而提高网络的整体性能。当网络流量减少时,同时减少系统开启线程数目,从而降低系统的功耗。

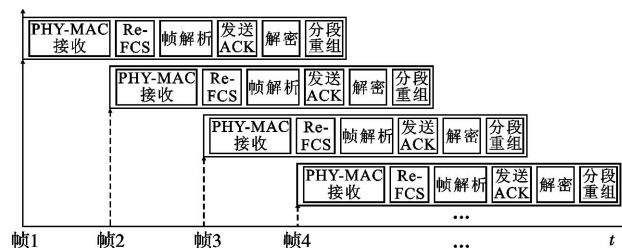


图3 基于同构多线程的软件MAC协议实现

4.2 基于网络实时性的异构多线程设计

根据 IEEE802.11 协议,源 STA 在发送 RTS/数据帧后等待一特定时间,若没有收到 CTS/ACK,则源 STA 断定 RTS/数据帧传输失败,进入差错恢复重传过程,这对 802.11 实时性提出了要求。而 802.11 中也有很多类型的帧不需要收到响应帧,如广播帧等,这是非实时性数据。如何根据网络实时性需求动态调整以使系统功耗最低是待解决的问题。

分析单线程 MAC 协议实现方法,它以一个 MAC 帧为单位,对 MAC 帧进行串行化的处理(图 4),先完整地接收一个 MAC 帧;然后进入重新 FCS 校验,如果 FCS 一致,则对 MAC 帧进行解析,然后进行其他相应处理,导致对 MAC 帧的处理需要较长时间,显然无法满足 802.11 的实时性需求。

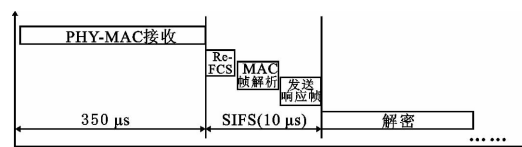


图4 单线程MAC协议接收数据处理过程

根据单线程 MAC 协议测试数据对比分析(如表 4、5 所示),表明随着 MAC 帧长度越来越大,对 MAC 帧的处理时间越来越多的花在 PHY-MAC 接收过程中。由于网络实时性的需求,其他处理过程必须在特定时间内完成,但 PHY-MAC 接收过程网络处理器需要处理的指令条数远远小于其余处理过程,网络处理器需执行的指令条数是不均衡的。

表4 802.11 MAC 协议数据接收处理时间及指令数

	MAC-PHY / PHY-MAC		其余接收处理过程 (无加解密)	
	时间/ μ s	指令条数	时间/ μ s	指令条数
RTS	6	6 374	<10 (SIFS)	7 299
CTS	5.5	6 246	<10 (SIFS)	7 306
DATA	350	55 155	<10 (SIFS)	33 522
ACK	5.6	6 250	<10 (SIFS)	6 866

表5 802.11 MAC 协议数据接收处理时间和指令数对比

	PHY-MAC		其余过程	
	时间比/%	指令数比/%	时间比/%	指令数比/%
RTS	37.50	46.60	62.50	53.40
CTS	35.50	46.10	64.50	53.90
DATA	97.20	62.20	2.80	37.80
ACK	35.90	47.70	64.10	52.30

本文通过分析 IEEE802.11 MAC 协议特点,提出以异构多

线程解决上述问题,主要思想是减少 MAC 协议处理粒度,将任务提前,以一个字节为单位,对一个字节处理完;然后将该字节送入下一步处理,从而提供以功能划分为基础的任务并行性。

基于异构多线程的软件 MAC 协议实现具体如图 5 所示。当 PHY 层向 MAC 层发送开始接收的信号时,开启 PHY-MAC 接收线程用于逐字节地从 PHY 接收 MAC 帧,当收到第一个字节的 MAC 帧后(图中 t_1),开启 FCS 重校验线程,开始对 MAC 帧 FCS 重校验。收到第二个字节的 MAC 帧后(图中 t_2),开启 MAC 帧解析线程,解析 MAC 帧类型,并解析 MAC 帧的目的地址并对帧进行过滤处理。根据 MAC 帧解析线程结果,如果是数据帧,则接收完 MAC 帧头开始接收第一个 MAC 帧实体时(图中 t_3),开启数据解密线程。在 MAC 帧解析线程完成后,如果目的地址无误且是数据帧(图中 t_4),开启构建 ACK 线程。随后直至 PHY-MAC 接收线程结束并将接收到的 FCS 与 FCS 重校验结果比较无误后(图中 t_5),开启发送线程完成协议控制,计算 ACK 的 FCS,并将 ACK 发送到 PHY 层。当数据解密完成后对 MAC 帧分段重组然后发送给 LLC 层。如果在某个线程中发现该 MAC 帧有误,应该立即通知其他线程做相应的错误处理。

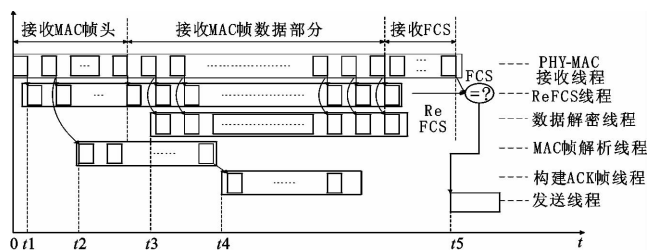


图5 基于异构多线程的软件MAC协议实现

由于 IEEE 802.11 MAC 协议各个任务的先后顺序是有结果依赖的,因此在异构多线程实现中,需要注意的是线程间的同步。比如 ACK 真正的发送必须在 FCS 校验无误后才能开始,帧解析必须是在 PHY-MAC 接收线程至少收到 MAC 帧头的前两个字节后开始等。

但是当 MAC 帧发生错误时,采用异构多线程将帧处理过程提前可能导致不必要的处理,从而增加网络处理器的工作量,甚至降低效率。所以该方法是以可能增加网络处理器的工作量为代价的,但 MAC 层收到错误帧的几率要远远小于收到正确帧的情况,对于绝大多数正确的 MAC 帧而言,该方法可以提高处理效率,最终达到网络实时性的需求。而针对网络中非实时性数据,可以动态调整线程数目,从而达到降低功耗的目的。

5 测试分析

5.1 基于网络流量的同构多线程设计

在假定物理层各项参数不变的情况下,继续以表 1 参数作为仿真环境,根据无线网络数据传输特点建模,在不同网络速率,不同数据加密方法下 MAC 层所需的最大同构多线程数如表 6 所示。

表 6 数据表明基于网络流量所需的 MAC 层同构多线程数目都在可接受范围内。因此在实际应用中,可以根据网络流量的大小动态调整系统开启的多线程数目,网络流量大时增加线程数目,网络流量减少时根据需要减少正在运行的线程数目,通过基于网络流量动态调整运行的线程数目从而达到系统功耗的最优化,如图 6 和 7 所示。

表 6 基于无线网络网络流量的 MAC 层最大同构多线程数目

网络速率/Mbps	最大网络流量/Mbps	加密方式	所需最大线程数
1	0.99	WEP-RC4	1
		CCMP-AES	1
2	1.95	WEP-RC4	1
		CCMP-AES	1
5.5	4.98	WEP-RC4	1
		CCMP-AES	2
11	9.01	WEP-RC4	1
		CCMP-AES	3
18	13.14	WEP-RC4	1
		CCMP-AES	4
24	16.01	WEP-RC4	1
		CCMP-AES	5
36	20.52	WEP-RC4	2
		CCMP-AES	7
54	25.24	WEP-RC4	2
		CCMP-AES	8
108	32.78	WEP-RC4	2
		CCMP-AES	10
300	40.52	WEP-RC4	3
		CCMP-AES	13
480	42.71	WEP-RC4	4
		CCMP-AES	14

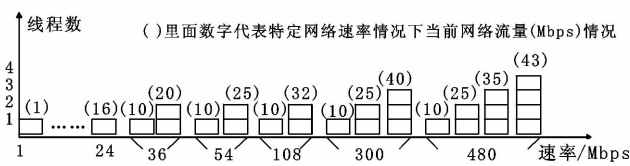


图6 网络流量自适应的协议实现(WEP-RC4)

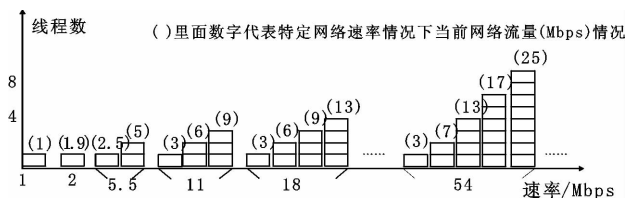


图7 网络流量自适应的协议实现(CCMP-AES)

5.2 基于网络实时性的异构多线程设计

MAC 帧分为许多不同的子类型,各个类型的 MAC 帧对实时性的需求不一,以最基本的 RTS/CTS/DATA/ACK 帧为例,RTS/DATA 帧为有实时性需求的 MAC 帧,而 CTS/ACK 则为无实时性需求的 MAC 帧。对于如帧解析、响应帧构建等线程,由于其指令较少,因此可以合并为一个线程串行执行。不同帧类型所需的异构线程数如表 7 所示,对于没有实时性需求的帧,只需两个硬件线程,一个用于接收 MAC 帧,另一个用于 MAC 帧解析;对于有实时性需求的帧,则需要更多的线程,在规定时间内完成后续处理,如图 5 所示 DATA 帧任务由五个软件线程构成,需要最多四个硬件线程支持(其中帧解析线程和构建 ACK 线程可以共享一个硬件线程)。测试表明可以通过 MAC 帧的实时性需求动态调整系统开启的多线程数目,从而达到系统功耗的最优化。

表 7 基于网络实时性需求的 MAC 层异构多线程数目

帧类型	是否要求实时性	所需最大线程数
RTS	是	3
CTS	否	2
DATA	是	4
ACK	否	2

6 结束语

本文基于硬件多线程网络处理器平台,通过构建

IEEE802.11 MAC 协议处理模型,对传统 MAC 协议实现方法通过测试数据研究发现其难以满足系统吞吐率和 MAC 协议实时性需求,进而提出一种基于硬件多线程的软件多线程 MAC 协议实现方法,使其能够根据流量和实时性自适应地调整系统的性能,从而降低处理器运行时的功耗,并通过测试分析验证了该方法的可行性。

参考文献:

- [1] 赵荣彩,唐志敏,邵洁. IXP 2400 网络处理器低功耗技术应用研究[J]. 计算机工程与应用, 2002, 38(22): 71-73.
- [2] YEONG J H, RAO X M, SHAJAN M R, *et al.* 802.11a MAC layer: firmware/hardware co-design[C]//Proc of 2003 Joint Conference of the 4th International Conference on Information, Communications and Signal Processing. 2003:1923-1928.
- [3] BUCAILLE I, TONNERRE A, OUVRY L, *et al.* MAC layer design for UWB LDR systems: PULSERS proposal[C]//Proc of the 4th Workshop on Positioning, Navigation and Communication. 2007:277-283.
- [4] FUJISAWA T, HASEGAWA J, TSUCHIE K, *et al.* A single-chip 802.11a MAC/PHY with a 32-b RISC processor[J]. IEEE Journal of Solid-State Circuits, 2003, 38(11): 2001-2009.
- [5] CHANG Chih-yung, SHIH Kuei-ping, LEE Shih-chieh, *et al.* On improving network connectivity by power-control and code-switching schemes for multihop packet radio networks[C]//Proc of 2005 International Conference on Wireless Networks, Communications and Mobile Computing. [S.l.]: IEEE, 2005: 540-545.
- [6] SARSHAR N, REZAEI B A, ROYCHOWDHURY V P. Low latency wireless Ad hoc networking: power and bandwidth challenges and a solution[J]. IEEE/ACM Transactions on Networking, 2008, 16(2): 335-346.
- [7] CHRISTENSEN K J, GUNARATNE C, NORDMAN B, *et al.* The next frontier for communications networks: power management[J]. Computer Communications, 2004, 27(18): 1758-1770.
- [8] LEE Seong-won, GAUDIOT Jean-luc. Throttling-based resource management in high performance multithreaded architectures[J]. IEEE Transactions on Computers, 2006, 55(9): 1142-1152.
- [9] AYERS J, MAYARAM K, FIEZ T S. Tradeoffs in the design of CMOS receivers for low power wireless sensor networks[C]//Proc of IEEE International Symposium on Circuits and Systems. 2007: 1345-1348.
- [10] YAP K S, BOEY K H. Clock gating methodology for high performance network processor in 90 nm[C]//Proc of IEEE International Conference on Semiconductor Electronics. [S.l.]: IEEE, 2004.
- [11] XIE Jian-yang, TANG Xiang-long, LI Tie-cai. low power design based on neural network forecasting for interconnection networks[C]//Proc of the 6th World Congress on Intelligent Control and Automation. 2006: 2979-2983.
- [12] TILLICH S, GROBSCHADL J. Instruction set extensions for efficient AES implementation on 32-bit processors[C]//Proc of the 8th International Conference on Cryptographic Hardware and Embedded Systems. Berlin: Springer-Verlag, 2006: 270-284.
- [13] WANG Qin, WANG Lei, QI Yue, *et al.* A hardware multi-threading architecture for protocol processors[C]//Proc of the 12th IEEE International Conference on Communication Technology. 2010.