

基于 μ clinux 的 EPA 通信协议栈^①

蒋焕军^{1,2}, 韩卫光²

1(中国科学院研究生院, 北京 100039)

2(中国科学院沈阳计算技术研究所, 沈阳 110168)

摘要: EPA 标准是由我国提出的用于提高工业以太网实时通信性能的系统规范。在介绍 EPA 通信协议模型基础上, 对通信协议栈的应用层和通信调度管理实体进行了模块化设计, 对协议栈的任务进行了详细划分。在设计过程中, 对 μ clinux 系统的硬实时性和内存利用率进行了重点考虑。经实验测试, 该通信协议栈可以正常工作, 并取得了较理想的效果。

关键词: 通信协议栈; 实时以太网; EPA; μ clinux; 通信调度管理实体

EPA Communication Protocol Stack Based on μ clinux

JIANG Huan-Jun^{1,2}, HAN Wei-Guang²

¹(Graduate University, Chinese Academy of Sciences, Beijing 100039, China)

²(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

Abstract: EPA Standard is a system specification that was proposed by China to improve industrial ethernet real-time communication performance. On the basis of introduction of the communications protocol model, the modular design of the application layer and the communications scheduling entity of the communications protocol is described, and the task of the protocol stack is divided in detailed. In the design process, the hard real-time and memory utilization of the μ clinux system are key consideration. The test results show that the communication protocol stack can work properly and it achieves better results.

Key words: communication protocol stack; real-time ethernet; EPA; μ clinux; communication scheduling management entity

在国家“863”计划支持下, 我国制定了第一个拥有自主知识产权的现场总线国家标准, 即 EPA(Ethernet for Plant Automation)标准。该标准适用于工业现场仪器仪表之间以及工业自动化控制系统和仪器仪表间的通信。它已被列入现场总线国际标准 IEC 61158 (第四版) 中的第十四类型, 并列为与 IEC 61158 相配套的实时以太网应用行规国际标准 IEC 61784-2 中的第十四应用行规簇^[1,2]。

μ clinux 是一种开源操作系统, 因其内核高效性和健壮性、有效网络通信支持能力、丰富的开放应用软件等优点, 已被广泛应用于嵌入式设备中。ARM9TDMI 是 32 位定点 RISC 处理器, 能够提供

1.1MIPS/MHz 的指令执行速度, 并具有代码密度高、功耗低、支持实时调试等优点。在本文中选用 μ clinux 操作系统和 ARM9TDMI 处理器作为 EPA 通信协议栈的实现平台。

1 EPA通信模型

EPA 通信模型采用了 ISO/OSI 通信参考模型的物理层、数据链路层、网络层、传输层、应用层^[3]。其中, 应用层除了使用 HTTP、FTP 等常用通信协议之外, 还加入了 EPA 应用协议。同时 EPA 通信模型在应用层之上增加了用户层(采用 IEC61499/61804 标准), 在网络层和 MAC 层之间添加了一个 EPA 通信调度管

① 基金项目:国家科技部科技支撑计划(2007BAP20B01)

收稿时间:2011-01-24;收到修改稿时间:2011-03-02

理实体 (Communication Scheduling Management Entity, 简称 EPA_CSME), 构成 6 层结构的通信模型, 如图 1 所示。

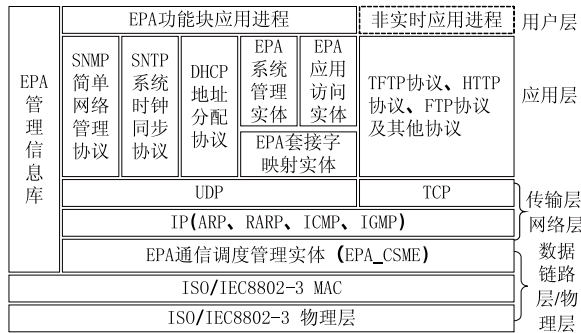


图 1 EPA 通信协议模型

2 通信模块设计

通信模块设计主要是在系统各模块基础上对服务数据处理程序的设计。依据 EPA 标准和 μ clinux 内嵌 TCP/IP 协议栈中已有的协议模块, 主要对 EPA 应用层各模块和 EPA 通信调度模块进行设计。

2.1 应用层设计

2.1.1 应用层模块划分

通过分析应用层结构和功能, 采用结构化设计思想, 应用层设计可分成六个模块^[4]: 系统管理服务模块、变量管理服务模块、域管理服务模块、事件管理服务模块、信息管理库操作模块和套接字映射实体模块。图 2 给出了各模块之间的关系。

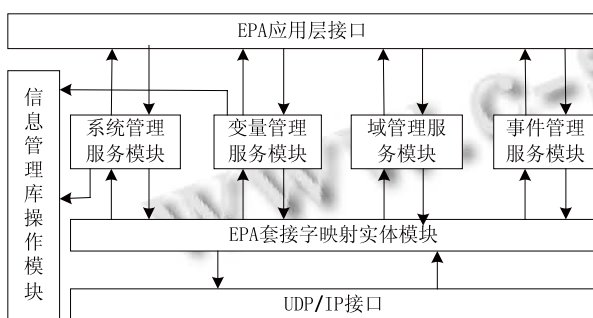


图 2 应用层各模块关系图

应用层各模块通过协同工作, 实现了 EPA 网络上设备之间的可互操作。

2.1.2 应用层内存管理

用户层的 EPA 功能块应用进程在发送数据之前, 需要根据服务标识, 来调用应用层的相应服务实体对

数据进行封装。进行封装前必须根据不同服务类型和数据大小先从内核申请内存, 若每次都通过 malloc() 和 free() 函数从内核的堆中申请和释放内存, 经过长时间, 堆中必然产生大量碎片, 从而降低内存利用率。同时, malloc() 和 free() 函数的多次执行也要占去大量 CPU 资源。

在此过程中, 可以通过池式内存管理来提高内存利用率和 CPU 运行效率。即在协议栈初始化期间, 预先从系统中申请不同尺寸的内存池, 而不同内存池通过单链表的方式进行组织。如图 3 所示, 结构体 A 用来描述一个内存池, 其中 size 记录该池中块的大小, first 指向池中第一个空闲块, next 指向下一个池的结构体, 而池的结构体按 size 大小组成链表。结构体 B 用来描述块, 其中, addr 指向一块大小为 size 的内存空间, next 指向下一个空闲块。当应用层的服务实体申请空间时, 从相应的池中取出第一个空闲块, 同时把结构体 A 的 first 设为下一个空闲块地址。某块被释放时挂到相应尺寸块链表结尾即可。

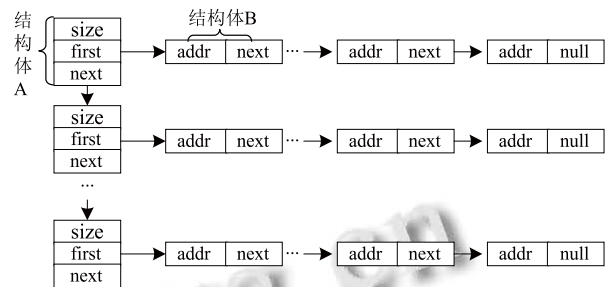


图 3 池式内存管理图

通过池式内存管理, 可避免动态内存分配引起的内存碎片, 可以使嵌入式设备上的宝贵内存得到充分利用。

2.2 通信调度模块设计

EPA 标准创造性地通过在网络层和数据链路层的 MAC 子层之间添加了一个 EPA 通信调度管理实体 (EPA_CSME), 解决了商用以太网通信不确定缺点。EPA_CSME 设计的好坏直接影响到整个协议栈的性能。图 4 给出了 EPA_CSME 通信调度的整体流程图。

2.2.1 数据链路层报文队列设计

EPA_CSME 收到从网络层 ip_finish-output() 传输过来的数据时, 并没有直接把数据发送到网络适配器上, 而是按报文类型进行缓存。在设计时需要构造 3 个缓存队列: 周期队列、非周期队列、声明队列, 分

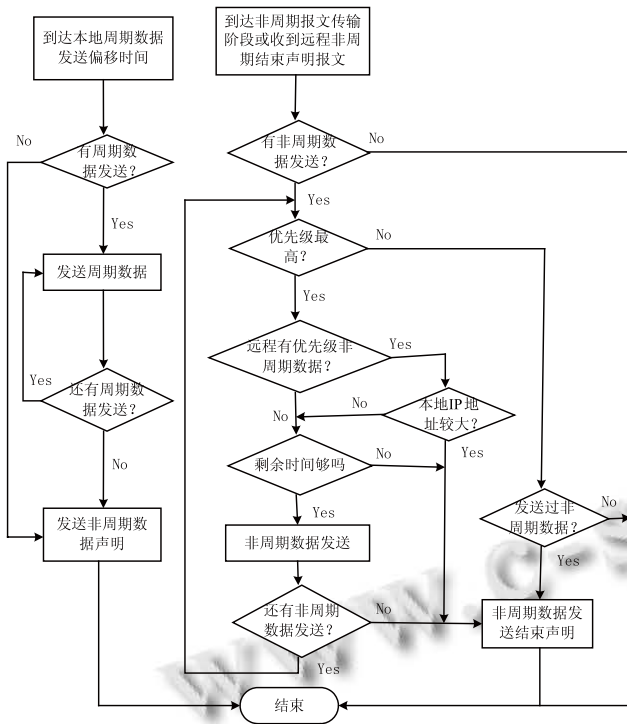


图 4 EPA_CSME 通信调度流程图

别用于存储 EPA 周期报文、EPA 非周期报文和远程 EPA 非周期声明/结束声明报文。以声明队列为例，队列节点的数据结构可定义为

```
typedef struct annnode
{
    struct annnode * next; //下一节点
    unsigned char pri; //优先级
    unsigned char haddr[6]; //MAC 地址
    unsigned char ip[4]; //IP 地址
}EpaAnnNode;
声明队列涉及到的操作函数主要有
unsigned char GetAnnPktPri(unsigned char * pkt);
//获取优先级
void EpaAnnListDelHead(EpaAnnNode * head);
//删除首节点
```

```
void EpaAnnAdNode(EpaAnnNode*head, EpaAnnNode * new); //按优先级插入新节点
```

2.2.2 调度设计

① 周期数据调度设计

当本地周期数据发送时间事件被触发时就会检测周期报文缓冲队列中是否有数据需要发送，如果有则将队列中周期数据依次发送到网路适配器上，最后发

送非周期数据声明报文。

本地周期数据发送时间事件触发机制可以通过 delta^[5]定时器定时来实现。当 EPA_CSME 启动时首先根据当前网络时间算出周期数据发送时间偏移量，并通过定时器进行定时。当本地周期数据发送时间到达时，首先算出下一周期数据发送时间偏移量并用定时器再一次定时，然后再发送周期数据。通过对本地周期数据发送时间循环定时，可以避免定时过程中出现误差的累计。

周期报文缓冲队列可以用环形数组进行描述，可以有效保证数据先进先出。

② 非周期数据调度设计

当非周期报文发送时间到来时或本地收到远程的非周期报文发送结束声明时，EPA_CSME 进入非周期报文调度状态。此时，如果声明队列的首节点的 MAC 地址和本地 MAC 地址相同，则把非周期队列中首个报文发送到网络适配器上，同时删除声明队列和非周期队列的首节点。

3 协议栈任务划分

3.1 提高系统实时性

本协议栈是在 μ clinux 系统下进行设计的，μ clinux 是 linux 的一个嵌入式变种。μ clinux 虽然采取基于优先级的调度策略，也将进程分为实时进程和非实时进程，但其采用的内核不可抢占和临界区操作时屏蔽中断机制，导致其很难完成硬实时任务。通过在 μ clinux 内核之外，以可加载内核模块形式添加一个实时内核 RTAI(Real Time Application Interface)，很好解决了通信协议栈中一些任务对系统硬实时性的要求。RTAI^[6]是一个完全的占先式内核，它接管了所有的硬件资源，将 μ clinux 内核作为它的一个低优先级任务来运行。在 RTAI/μ clinux 双内核结构下，协议栈实时性任务在 RTAI 的调度下运行，非实时性任务在 μ clinux 调度下运行。

3.2 任务划分

模块和任务非一一对应关系。模块是基于功能进行划分，而任务是基于时间优先级进行划分。在设计时，根据任务的实时性要求来分配任务的优先级，实时性要求高的任务优先级高，实时性要求低的任务优先级低。对协议栈的任务划分按优先级由高到低可以划分为以下几个：

周期报文发送任务：当本地周期报文发送时间到来时，定时器发送信号唤醒该任务。周期报文发送给网络适配器以后，该任务立即挂起。

非周期报文发送任务：当一个通信周期中的非周期传输阶段到来时，定时器发送信号唤醒该任务。该任务判断本地待发送的非周期报文优先级是否最高，若是，发送非周期报文到网络适配器，否则，该任务立即挂起。由于该任务和周期报文发送任务不会在同一时间段执行，任务级别可以设置相同。

定时器任务：对定时器链表进行维护，当某任务执行时间到来时，给相应任务发送信号。

非周期报文优先级维护任务：当网络适配器收到非周期声明报文或非周期数据发送结束声明报文时，激活该任务。该任务对非周期数据调度设计一节所述的声明队列和非周期队列进行维护。如果当前处于非周期报文传输时间段且声明队列的首节点的 MAC 地址和本地 MAC 地址相同，则该任务会发送信号唤醒非周期报文发送任务。

发送队列处理任务：该任务取出应用层中优先级最高的发送队列中的报文进行发送。当发送队列均为空时阻塞该任务，只要有一个发送队列变满时就唤醒该任务。在发送报文时，对于无证实报文，直接发送即可，而对于证实服务请求报文，需要在定时器链表中添加一个与报文标识相关的定时器，以便于对超时报文进行重发。

接收队列处理任务：该任务将应用层接收队列中的首个 EPA 报文发送给应用层服务实体进行处理。当应用层服务实体收到期待的响应报文时，会将定时器链表中的相应定时器删除。该任务会在接收队列为空时挂起，在接收队列由空变满时唤醒。

接收缓冲区处理任务：该任务不断从 UDP 数据包缓冲区中取出数据包，若数据包是 EPA 报文，则送给 EPA 报文解析功能块进行解析，并将解析出的 EPA 包缓存到接收队列中。对于非 EPA 报文数据包直接丢弃即可。

图 5 为通信过程信息处理流程。应用层缓存队列部分，针对三种粒度不同内存池，设计了三个不同优先级的发送队列。通信调度层针对三种 EPA 报文类型设计了三个缓存队列。

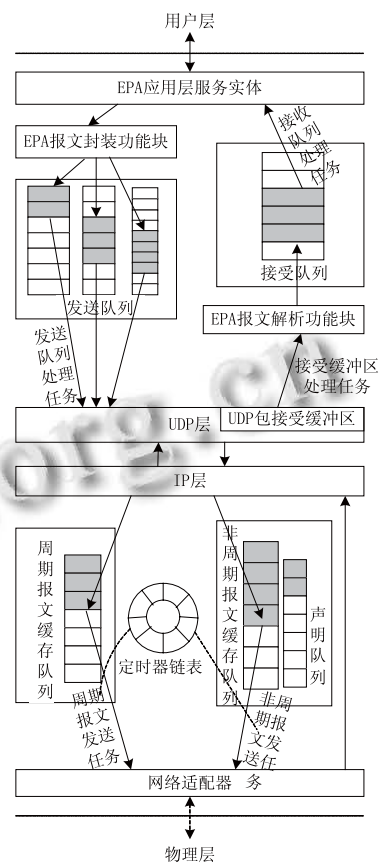


图 5 协议栈信息处理流程图

4 实验和结果分析

测试 EPA 通信协议栈性能常需编写专业软件，工作量很大。而通过以下给出的实验方案可以简单测试协议栈通信性能。

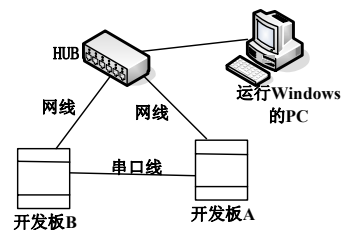


图 6 试验平台拓扑结构

实验方案设计：试验拓扑结构如图 6 所示，装有 EPA 通信协议栈的开发板 A 接受信号发生器产生的模拟信号，将接受的模拟信号转换成数字信号上传给开发板 B，同时也用串口将数字信号上传给开发板 B。开发板 B 对两次传递的信号进行比较和分析并进行存

储。运行 Windows 的上位机通过 CGI 技术以网页的形式从开发板 B 上读出比较结果。

结果分析:图 7 中(a)图是当 Windows 机从开发板 B 读出的比较结果。横轴为时间轴,单位为秒,纵轴为接受数据量,单位为千字节。根据显示的曲线结果,可以看出通过网络传输的趋势曲线与通过串口传输的趋势曲线是一致的。假设串口传递的数据是实时的且没有数据包丢失,而通过协议栈网络传递的数据是非实时的或在传递过程中发生数据包的丢失,则产生的曲线不会吻合,会产生时移。实验结果所显示的两条曲线完全的一致表明协议栈可以实时的把数据传输到网络上且没有数据包丢失。当设置串口在传输数据

前先缓冲数据 20ms,然后再发送数据时,可以得到图(b)比较结果,从图中可以看出网络传输实时性高于串口传输,但两条曲线走势完全相同,说明传输过程没有丢失数据包。

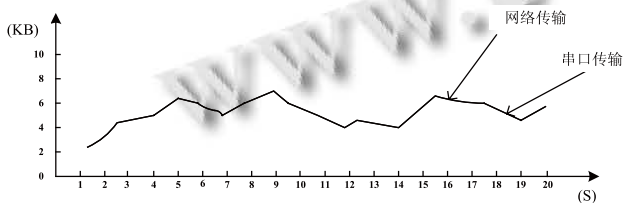
5 结语

本协议栈是在 μ clinux 系统内嵌的 TCP/IP 协议栈各功能模块基础上进行设计的。网络适配器用的是 DM9000E 芯片,设计过程中,对 dm9000.c 驱动程序中的接收包中断处理函数进行了修改,以满足协议栈性能需要。考虑到 μ clinux 对 C 支持性比较好,对设计的模块全部用标准 C 进行开发。最后通过设计的实验平台对 EPA 通信协议栈性能进行测试,测试结果显示该 EPA 通信协议栈通信过程稳定,速度快。

参考文献

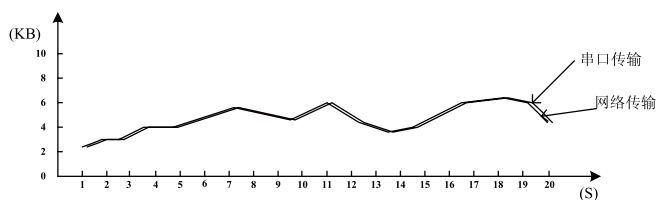
- 1 自动化在线网站.[2010-12-16]. <http://www.autooo.net/autooo/wuxiantongxun/Nesw/2008-01-16/47899.html>
- 2 陈磊,冯冬芹,金建祥,褚健.以太网在工业应用中的实时特性研究.浙江大学学报:工学版,2004,38(6):670-675.
- 3 国家质量技术监督局.GB/T20171-2006.用于工业测量与控制系统的 EPA 系统结构与通信规范.北京:中国标准出版社,2006.
- 4 张鑫龙,王平,孙攀.EPA 通信协议栈的设计与实现.重庆邮电学院学报:自然科学版,2006,18(4):512-515.
- 5 王兰,梁杰申.基于 EPA 的实时通信方案.机电工程技术,2008,37(5):49-50.
- 6 温现杰.基于 Linux 内核的实时调度机制的研究和实现.哈尔滨:哈尔滨理工大学,2009.

网络传输和串口传输数据趋势曲线



(a)

网络传输和串口传输数据趋势曲线



(b)

图 7 实验结果

(上接第 240 页)

的 Coinbox 类软件测试中。该方法证实了可以很好地描述自动售货机的 Coinbox 类的动态转换行为,生成的测试用例可以发现动态行为转换中的错误,便于测试用例生成和测试实现。但该方法仍存在很多不足,它不能对自动售货机的 Coinbox 类组合中存在的一些状态切换等问题进行很好的测试,生成的测试用例也还不能很好地优化。

参考文献

- 1 Dssouli R, Saleh K, Aboulhamid E, et al. Test Development for Communication Protocols: Towards Automation. Computer

Networks, 1999,31(17):1835-1872.

- 2 年晓玲.基于扩展有限状态机软件测试用例自动生成的研究.西南交通大学,2005.24-44.
- 3 Ammann P, Offutt J. Introduction To Software Testing. New York:Cambridge University Press, 2008.
- 4 Kung D, Suchak N, Hsia P, et al.On Object State Testing. Proc of COMPSAC'94. 1994.222-227.
- 5 卢炎生,雷波,谢晓东,毛澄映.基于扩展有限状态机的定义清除路径的生成方法.计算机工程与科学,2006,28(7):67-69.