

基于分布式路由器的 PPP 虚拟驱动

林 林, 张兴明, 李 丹, 申 涓

(解放军信息工程大学信息工程学院, 郑州 450002)

摘要: PPP 协议是一种广泛使用的链路层协议, 协议流程简洁且具有认证功能。该文结合高性能路由器项目, 提出一种基于虚拟驱动适配层的 PPP 协议虚拟驱动模块的设计方案。该方案能够为上层软件提供良好底层硬件设备的模拟和屏蔽, 满足与软硬件无关的设计思想, 提高系统的可扩展性和可维护性。经过测试验证, 该方案高效稳定, 易于维护和移植。

关键词: 分布式结构; 虚拟驱动适配层; PPP 协议; 虚拟驱动模块

PPP Virtual Drive Based on Distributed Routers

LIN Lin, ZHANG Xing-ming, LI Dan, SHEN Juan

(College of Information Engineering, PLA Information Engineering University, Zhengzhou 450002)

【Abstract】 Point to Point Protocol(PPP) is a popular link layer protocol with the characteristic of high efficiency and authentication. This article offers a design based on virtual drive adaptive layer that focuses on the distributed designation of high-performance router. And it focuses on the realization of the PPP virtual drive module in RAL framework. The module offers the upper software well shield and simulation of the devices. So it satisfies the software-hardware independent design, improves the extension and maintenance of the system. Experimental result shows that the scheme is highly effective and stable. And it is easy for maintenance and migration.

【Key words】 distributed architecture; virtual drive adaptive layer; Point to Point Protocol(PPP); virtual drive module

目前, 大多数嵌入式系统的研发采用与软硬件无关的分离设计方案, 以减小研发难度、缩短开发周期、增强软件系统对硬件设备升级的支持。为此, 本文提出在 Linux 操作系统环境下的虚拟驱动设计方案。虚拟驱动是指在操作系统内核中模拟一种设备及其行为, 并为上层软件提供统一的调用接口。虚拟驱动可以对应真实的硬件设备, 也可只对应一个软件过程。该设备的使用者只需通过调用虚拟驱动所提供的接口即可, 实际的数据传输和处理由虚拟驱动完成。采用对硬件抽象的虚拟驱动方案, 可以为上层提供良好的接口并屏蔽下层硬件设备的具体实现, 能够很好地满足与软硬件无关的设计思想, 保证其适应上层软件对下层硬件的变化。

1 路由器虚拟驱动总体设计

1.1 虚拟驱动设计思路及框架

作为骨干网的核心路由器, 高性能路由器硬件结构复杂, 具有接口数量大、种类多的特点。为提高系统性能, 实际的路由器总体设计方案采用了当前高端路由器设计中比较流行的集中式交换与协议处理, 分布式的数据报文处理与转发的分布式结构, 如图 1 所示。路由与转发分离的思想是基于“两级结构”的。其中, 主处理单元负责实现设备管理、路由计算等功能, 并将转发表下发到从处理单元。从处理单元根据主处理单元下发的转发表进行独立路由转发。基于如图 1 的分布式体系结构, 如何设计一种可扩展的且不依赖于硬件具体实现方式的软件体系结构, 构建通用性强的路由器基础平台, 方便商用路由软件的移植, 成为关键。

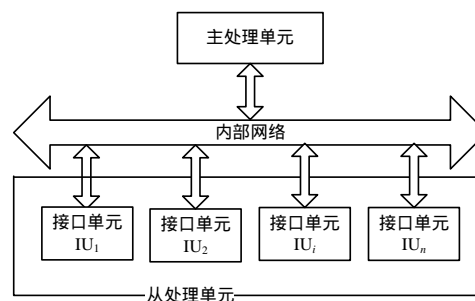


图 1 分布式结构

在一般的嵌入式系统中, 作为软硬件交互桥梁的设备驱动程序就可以提供这样一种平台, 来分离应用软件和底层硬件实现。但是在高性能路由器中, 情况比较复杂, 原因如下:

(1) 主处理单元和从处理单元是独立运行的, 即主处理单元上的操作系统并不知道各个分布式从处理单元上的接口, 它只能对自己所属的网络接口进行封装和发送, 因此, 需要有专用的模块将这些从处理单元上的接口映射到主处理单元中供其调用。

基金项目: 国家“863”计划基金资助项目“大规模接入汇聚路由器(ACR)系统性能与关键技术研究”(2004AA103130)

作者简介: 林 林(1981-), 男, 硕士研究生, 主研方向: IP 宽带网络, 网络拥塞控制; 张兴明, 教授; 李 丹, 博士研究生; 申 涓, 讲师、硕士

收稿日期: 2007-10-08 **E-mail:** priestath@hotmail.com

(2)一般的设备驱动程序都针对具体硬件设备的实现,而不具备通用性。当硬件发生变化时,驱动程序以及上层软件的设计都要做相应的调整,不能做到软硬件之间的透明设计。对于分布式路由器这类底层设备较多的系统,传统的驱动程序模式将限制系统的可扩展性和硬件设备的更新。

本文采用虚拟驱动的方式来为上层应用软件模拟下层设备,使上层协议软件通过虚拟驱动模拟的虚拟设备透明地操作各个分布式接口。同时,当下层硬件发生变化时,只需调整硬件设备驱动与虚拟驱动的接口,而使下层硬件的改变对上层软件透明。因此,提出如下设计思路:

(1)系统采用分布式体系结构,虚拟驱动的主要作用就是为上层协议软件模拟实际的底层物理设备,所有的物理接口在上层软件看来都是由虚拟驱动提供的虚拟接口单元。因此,虚拟驱动主要被设计用来为上层软件提供设备调用接口。

(2)除了为上层协议软件提供接口之外,虚拟驱动还需要和下层物理接口进行映射并通信,这样上下行的数据包才能真正地被投递到目的地。本文采用链路复用的思想,设计专用的内部通信模块来负责处理各个虚拟接口同物理设备间的数据传递,使之成为联系虚拟驱动和真实物理设备的通道。

本文提出一种虚拟驱动适配层的软件结构(见图2),作为实现虚拟驱动的一种普适性架构。该层包括2个部分:

(1)内部通信模块

处于虚拟驱动适配层的底层,它负责路由器内部网络的通信。

(2)虚拟驱动模块

位于内部通信模块之上,模拟底层硬件的行为。

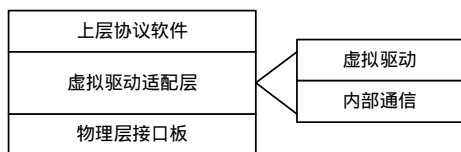


图2 虚拟驱动适配层的软件结构

虚拟驱动适配层作为一个独立的子系统,通过虚拟驱动和内部通信两模块之间的协同动作,映射从处理单元的硬件实现,使上层软件与硬件设备相分离。其中,虚拟驱动模块模拟了底层接口,支持上层应用的调用。同时,专用的内部通信模块保证了软硬件系统间信息交互的实时可靠。

1.2 虚拟驱动模块结构与功能

作为模拟下层真实接口的虚拟驱动模块,有其自身的结构特点:

(1)由于实际接口数量较多,虚拟设备采取调用时注册,链式队列管理的方式,按需分配并及时注销^[1]。

(2)为支持上层软件调用,虚拟驱动程序采取统一的用户调用接口和管理模式。同时,虚拟驱动使用不同设备编号来分别映射不同物理设备,从而实现虚拟设备与物理接口的一一对应。还有,虚拟驱动需要设计与内部通信模块的接口,以完成数据的收发。

虚拟驱动的功能是模拟从处理单元上的接口单元(Interface Unit, IU)。为了较为透明地实现这一功能,虚拟驱动采取网络接口驱动程序的方式,对上层完全采用标准的驱动程序接口,实现了与上层软件的无缝结合。

在虚拟驱动中,用来模拟实际接口的虚拟接口单元(Virtual Interface Unit, VIU)与IU的对应关系如图3所示。协

议软件对不同VIU所进行的操作,会通过虚拟驱动程序的处理,经由内部通信反映到从处理单元的IU上。反之,IU的状态变化以及收包的动作,也会通过虚拟驱动反映到VIU上,从而触发协议软件的一系列变化。

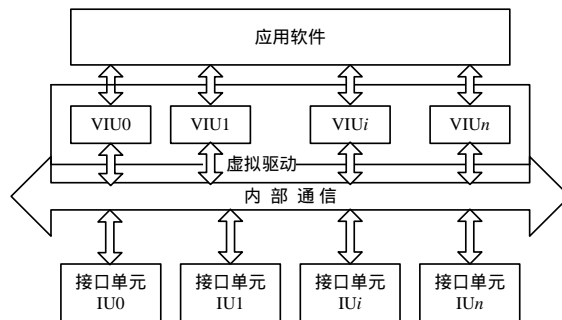


图3 虚拟接口单元与IU的对应关系

虚拟驱动与真实网络接口驱动的唯一区别,主要在于收发包的处理方式上。虚拟驱动模块的下层是内部通信模块,而非真正的硬件设备,所以,它的收发动作并不是对真实的网络接口卡的操作,而是与内部通信模块之间的数据传递。

对于上层协议软件的开发者来说,只需要了解应该使用什么样的接口来和下层硬件通信即可,而不需要了解硬件及其驱动程序的具体实现。当有新的上层应用软件需要进行移植时,只需要满足接口的匹配,从而避免了大量代码的改写,减小了移植的难度。相应地,当硬件部分发生变化时,其影响只涉及虚拟驱动与底层设备间的接口,只要这两者之间可以做到匹配,上层软件仍然可以正常地运行,保证了硬件实现对上层软件设计的透明性。

虚拟驱动为系统的软硬件之间提供了一个良好的屏障,带来的好处如下:

(1)软件开发与硬件设计可以同步进行,缩短开发周期;

(2)软硬件的分离,更加有利于上层协议软件的移植和下层硬件实现的更新;

(3)模块化的设计便于开发过程中的调试和使用过程中的维护。

在具体实现过程中,遵循虚拟驱动总体设计的普适性框架即虚拟驱动适配层结构的需求,并根据实际网络接口和链路的特点,选择具体协议来实现虚拟驱动模块。本文将针对采用PPP协议的虚拟驱动的设计与实现进行介绍和讨论。

2 PPP虚拟驱动的设计与实现

2.1 PPP软件模块的划分

PPP协议是一个协议族,由3类协议组成:链路控制协议族(LCP),网络层控制协议族(NCP),PPP扩展协议族^[2-3]。此外,PPP还提供用于网络安全方面的验证协议族(PAP,CHAP)。

为实现认证、协商和建立连接等一系列过程,PPP驱动模块不同于一般设备驱动只要向内核注册设备结构就可以运行,还需要有负责处理协议流程的部分。同时,按照PPP协议规定,协议软件需要在运行过程中随时接收来自内核或者用户的消息和命令,因此,还存在与用户的交互需求。鉴于此,将PPP虚拟驱动划分为2个模块,PPPD守护进程子模块和PPP虚拟驱动接口子模块。

PPPD即PPP Deamon子模块,由于需要随时准备进行链路配置或相应连接请求,同时还要与用户进行交互,因此采

取守护进程的方式部署于用户空间。在实际网络通信进行收发之前,执行全部的初始化工作和认证,包括:管理和控制各虚拟通信设备,处理各种 PPP 控制协议,运行协议状态机以及在运行过程中随时响应用户的命令。

PPP 虚拟驱动接口子模块主要功能是向内核注册设备模拟实际物理接口,并向上层软件(PPPD 及协议软件)提供收发过程的系统调用接口,同时,完成与内部通信间的数据交互。部署于 linux 操作系统的内核,负责封装解封装、接收和发送报文。

2.2 PPPD 子模块设计

作为 PPP 协议实现的主要部分,PPPD 实现该协议的逻辑功能,即 PPP 协议自动机^[4-5](见图 4)。

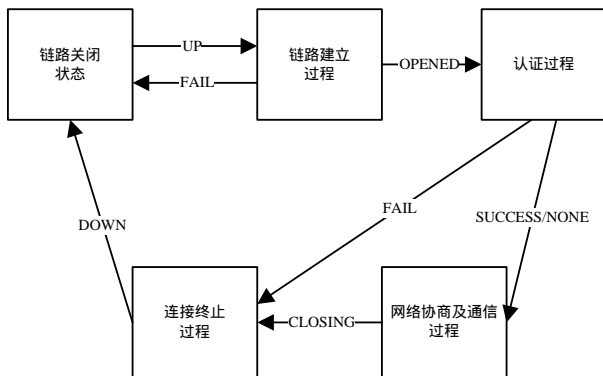


图 4 PPPD 子模块主要设计流程

针对上述自动机流程,PPPD 子模块需要完成如下功能:

- (1)进行 PPP 网络接口设备的创建注册和注销;
- (2)进行 PPP 字符设备的打开和关闭;
- (3)进行 PPP 网络接口设备中发送队列节点的创建和销毁;
- (4)进行数据链路的建立、配置和探测连接;
- (5)进行 PPP 认证(如果 LCP 协商要求认证);
- (6)进行各网络层协议的建立和配置;
- (7)数据链路的维护和终止。

2.3 PPP 虚拟驱动接口子模块设计

相对于 PPPD 模块实现协议自动机功能,PPP 虚拟驱动接口子模块主要为 PPPD 模块以及上层协议栈软件提供对真实硬件设备的模拟,并实现与内部通信间的接口。

需要完成的功能包括:

- (1)对发往内部通信模块的包进行封装,并对从内部通信接收包的解封装;
- (2)给协议栈提供标准的收发包函数;
- (3)给 PPPD 提供标准的收发包函数;
- (4)给内部通信提供 ioctl()系统调用的接口函数,用于同内部通信收发包;
- (5)给 PPPD 提供 ioctl()系统调用的接口函数,用于 PPPD 进行设备初始化的接口。

PPP 虚拟驱动接口子模块包括 ppp_generic, ppp_deflate, bsd_comp, slhc 4 个内核模块。其中,ppp_generic 是最主要的

模块,它提供了内部通信、协议栈和 PPPD 的接口函数,负责实现这三者之间的数据流通。ppp_deflate 和 bsd_comp 模块负责收发包的压缩和解压缩。

2.4 PPP 虚拟驱动数据流程

经过实验测试,前述方案可以比较好地完成基于 PPP 协议的虚拟驱动设计,实现协议自动机的运行和正常的数据包收发。

当物理接口板从路由器外部的点对点链路上收到 PPP 包时,就经由内部通信递交到位于主控板的 PPP 虚拟驱动接口。然后 PPP 虚拟驱动接口分析包类型,分为 PPP 控制包和网络包。如果是 PPP 控制包,则将该包加入 PPP 网络接口设备的接收队列,然后 PPPD 子模块从中逐一读取进行处理。如果是网络包,而且链路协商、网络协商和认证阶段都已通过,则递交给协议栈处理。反之,如果协商过程或者认证过程出现错误,自动机将丢弃收到的数据包而不予处理或通告。

发送过程与之相反,当 PPPD 子模块或者协议栈要发送 PPP 控制包或网络包,首先调用 PPP 虚拟驱动接口模块提供的发送接口函数,将包插入 PPP 网络接口设备发送队列的末尾,然后逐一出队,调用 PPP 网络接口设备的发送函数发送,经由内部通信递交给物理接口板,然后由接口板发送出去。图 5 显示了内部通信、PPPD、PPP 虚拟驱动接口和协议栈之间的数据通路。

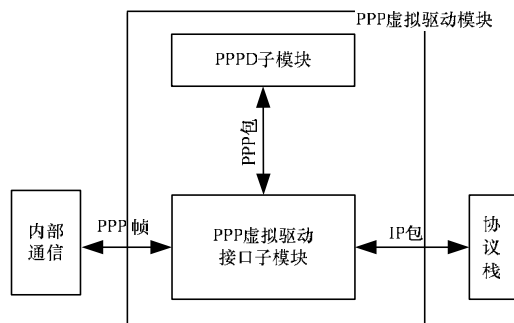


图 5 内部通信、PPP 虚拟驱动模块和协议栈之间的数据通路

3 结束语

本文结合高性能路由器项目提出了一种针对分布式结构路由器的 PPP 虚拟驱动设计方案。该方案有针对性地解决了如何实现下层硬件对上层软件透明的问题,提高了系统的可扩展性和可维护性,对于分布式结构的路由器系统设计具有较好的普适性。

参考文献

- [1] Corbet J, Rubini A. Linux 设备驱动程序[M]. 北京: 中国电力出版社, 2006-02.
- [2] Simpson W. The Point-to-Point Protocol[S]. RFC1661, 1994-07.
- [3] Rand D. PPP Reliable Transmission[S]. RFC1663, 1994-07.
- [4] McGregor G. The PPP Internet Protocol Control Protocol (IPCP)[S]. RFC1332, 1992-05.
- [5] 陆魁军, 史 烈. 基于路由器的 PPP 协议关键技术研究[J]. 计算机工程, 1999, 25(3): 3.