

基于 RNDIS 协议的虚拟网卡程序实现

高东飞, 张志鸿

(郑州大学 信息工程学院, 河南 郑州 450000)

摘要:微软提出的 RNDIS 协议使得以太网通信能够在 USB 总线上传输,而微软在实现 RNDIS 时使用 TCP 协议,通信效率不高。为略去 TCP 三次握手过程,提高 RNDIS 协议的通信效率,论文在 USB2.0 协议和 RNDIS 协议的基础上,介绍了 RNDIS 协议的结构及其在 USB 网络设备上的应用,分析了 RNDIS 协议的通信流程,同时结合 USB 设备主机,设计并实现了基于 Windows Mobile 设备的虚拟网卡程序。该程序在 LIBUSB 无驱设计的基础上,实现了基于 UDP 协议的 RNDIS 协议的方法,通过测试证明基于 UDP 协议的 RNDIS 协议方法比 TCP 方式的通信效率更高。

关键词:USB 协议;RNDIS 协议;LIBUSB 技术;Windows Mobile 设备;虚拟网卡

中图分类号:TP334

文献标识码:A

文章编号:1673-629X(2013)02-0085-04

doi:10.3969/j.issn.1673-2013.02.019

Realization of Virtual Network Card Program Based on RNDIS Protocol

GAO Dong-fei, ZHANG Zhi-hong

(School of Information Engineering, Zhengzhou University, Zhengzhou 450000, China)

Abstract:Microsoft's RNDIS protocol enables Ethernet communication to transmit over the USB bus. And Microsoft in the realization of RNDIS using the TCP protocol, the communication efficiency is not high. To remove the TCP three-way handshake process to improve the communication efficiency of the RNDIS protocol, it describes the RNDIS protocol structures and their application on networking equipment, analyzed the RNDIS protocol communication processes based on USB 2.0 and RNDIS protocol. Combined with host USB devices, design and implement virtual card procedures based on Windows Mobile. On the basis of no drive in LIBUSB, the program implements the RNDIS protocol based on UDP protocol. Tests proved the RNDIS protocol based on UDP protocol method is more efficient than TCP communications.

Key words:USB protocol;RNDIS protocol;LIBUSB technology;Windows Mobile device;virtual network card

0 引言

随着 USB 技术的快速发展及广泛应用,基于 USB 总线上的网络设备在许多应用领域代替了传统的 ISA、PCI 网卡,特别在便携式领域(如 iPad、智能手机等)和家庭网络领域中基于 USB 接口与主机进行类似于以太网的通信占据主导地位。为了实现基于 USB 总线进行以太网协议的通信,微软于 2002 年提出了 REMOTE_NDIS(RNDIS)规范,论文实现了基于该规范的 Windows Mobile 设备虚拟网卡程序,该程序比微软的 ActiveSync 同步软件的数据传输更高效。

1 RNDIS 的基本原理

RNDIS^[1]是微软为动态的即插即用型总线(如 USB, IEEE1394, Bluetooth, InfiniBand 等)网络设备提出的一套协议规范,此规范定义了一套独立的消息协议及消息在主机和 RNDIS 设备之间传送的方式,并在总线映射中定义了特定总线上的特定功能,RNDIS 是相当简单的易于理解和行之有效的 NDIS 体系结构的扩展,RNDIS 通过定义消息来封装 NDIS 微型端口接口,将 NDIS 的微型端口驱动程序处理代码移入设备本身。

1.1 NDIS 的架构

NDIS 是微软 Windows 操作系统中使用的网络体系结构的一部分,NDIS^[2]提供了简化的微型端口设备驱动程序体系结构,以便与使用常见驱动程序接口的网络适配器进行通信的传输,图 1 显示了一个传输(协议驱动程序),NDIS 和网卡微型端口设备驱动程序之间的关系。显示了连接使用了传统的 PCI I/O 总

收稿日期:2012-05-25;修回日期:2012-08-27

基金项目:国家科技重大专项项目(2009ZX0300101801)

作者简介:高东飞(1988-),男,河南新乡人,硕士研究生,研究方向为智能信息处理;张志鸿,博士,教授,中国计算机学会高级会员,主要从事分布式系统、网络计算技术、企业信息化工程等方面的研究与开发工作。

线网络接口卡。

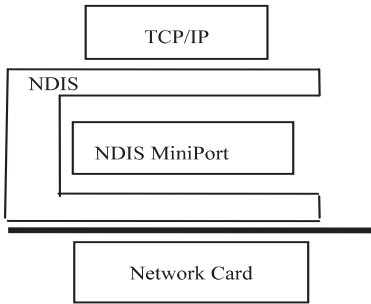


图 1 NDIS 结构模型

1.2 早期串行总线网络设备的解决方案

RNDIS 出现之前,串行总线网络设备的制造商必须为他们的设备定制 NDIS 微端口驱动程序,该驱动程序同时还与串行总线 WDM 驱动程序连接,不同的串行总线必须开发不同的驱动,这无形中加重了制造商的负担。其解决方案如图 2 所示:

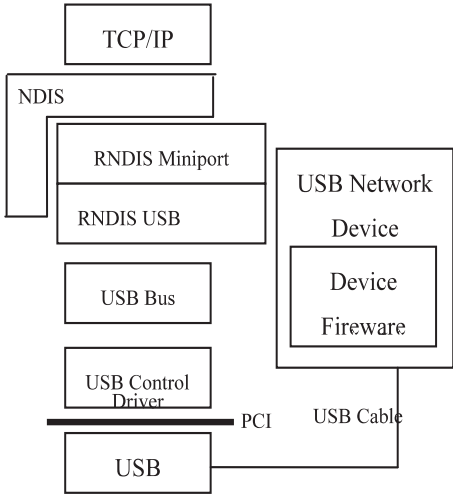


图 2 NDIS 设备在网络上的应用

图 2 中的“NDIS Miniport”就是设备制造商所必须去扩展的 NDIS 驱动部分,此驱动程序是专门为其设备和 USB 驱动程序的接口而写的。

1.3 RNDIS 方案

RNDIS^[3]消除了硬件供应商通过定义独立总线的消息集和此消息集在各种 I/O 总线上运行方式的说明,而编写一个 NDIS 微型端口设备驱动程序的需要,因为此 RNDIS 接口是标准化的,一组主机驱动程序可以支持任何数量总线上的网络设备。提高系统的整体稳定性,使设备制造商省去了编写微端口驱动的麻烦,用户也不需要安装支持新连接总线网络设备的驱动程序(这些微端口驱动将由微软统一提供)。该体系结构的映射通过创建 RNDIS 微型端口驱动程序实现 RNDIS 消息组,通用总线传输驱动程序依次与相应的总线驱动程序进行通信。

RNDIS 消息封装了 NDIS 微端口驱动的一些接口定义,把本来由 NDIS 微端口驱动需要处理的代码交

到设备本身。RNDIS 消息设置^[4]如下:

- (1)设备的初始化、挂起、复位等操作消息;
- (2)发送和接收数据包;
- (3)设置和询问设备的有关参数(如通过设置可以从设备初始状态进入数据初始状态,查询 mac 地址,最大包的字节数等);
- (4)指示媒体链接状态和监视设备状态。

对于每个支持 RNDIS 的串行总线类型提供总线传输驱动程序,用于实现载送 RNDIS 的消息,这类驱动功能有限,它只负责在 RNDIS 微端口驱动和特定串行总线驱动程序(如 USB)间传递 RNDIS 的消息,方案的结构模型如图 3 所示。

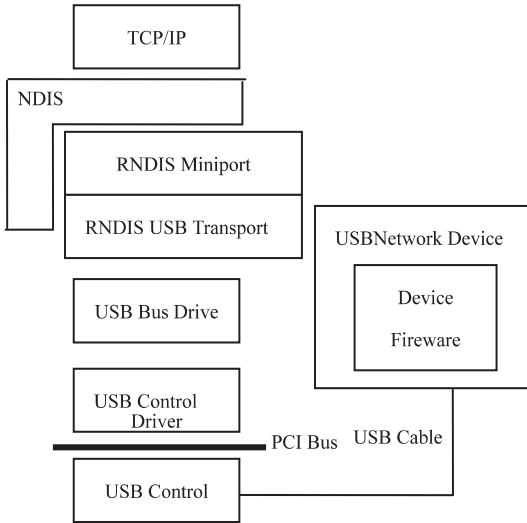


图 3 RNDIS 设备在网络上的应用

由图 3 可知设备制造商的 NDIS 微型端口被 RNDIS 微型端口驱动程序 USB 传输共同代替,设备制造商将不必关心驱动程序部分。

2 设备端程序的设计

设备端的程序是基于 libusb 来实现的,应用程序通过 libusb^[5]调用外部 API 来操作硬件,Libusb-Win32 是 Windows 操作系统上的通用 USB 设备驱动程序^[6]。Libusb 可以让用户不需要编写驱动程序就可以调用操作系统上的 USB 设备。

Libusb 具有以下特点:首先,能够与任意一个已安装的 USB 设备进行通信;其次,libusb 可以被用作自己开发的 USB 设备的驱动程序;再次,支持批量和中断传输、支持 USB 规范中定义的所有标准设备请求;最后,支持 USB 设备制造商的自定义请求。

设备端的程序设计在结构上必须符合微软提供的 RNDIS 的规范,同时还要遵循 USB 2.0 协议。

2.1 USB 设备描述

根据 RNDIS 规范,接入主机上的 USB 设备将被配置为 4 个 USB 端点^[7],端点 0,两个 BULK 端点(usb_

bulk_in 为端点 2,usb_bulk_out 为端点 3) 和一个中断 IN 端点(端点 1),其中设备的通信接口由默认端点(端点 0)和中断端点(端点 1)组成,而 usb_bulk_in 端点(端点 2)和 usb_bulk_out 端点(端点 3)组成了设备的数据接口,这两个接口的集合就是该设备的配置;同时该设备类型属于通信类设备,该通信类接口中的子类是抽象控制模型,该模型^[8] CDC 协议,通过定义两个常量数组来分别描述该类设备的设备描述和配置描述符。

2.2 程序的总体结构

该程序的实现分为两部分,主程序和 Generic RNDIS 端的 SOCKET 程序,主程序部分主要完成硬件抽象层的初始化,如 Generic RNDIS 设备的初始化和配置等等;Generic RNDIS 设备端的程序完成数据的收发功能。

2.3 RNDIS 协议的实现

设备的配置过程主要是针对 USB 主机^[9]发向 Generic RNDIS 设备的请求进行处理的,有以下几个阶段:

- (1)总线复位;
- (2)主机读取设备描述符;
- (3)地址资源的分配;
- (4)主机从新的地址资源中读取 PID、VID 及描述符集合。

一个没有驱动的 Generic RNDIS 设备和 USB 主机的通信过程到这里就结束了。接下来就是本程序要实现的功能了。这时 USB 主机端程序首先会发送一个 48 Bytes 的 RNDIS_INIT_MSG 报文给 Generic RNDIS 设备的控制端点(端点 0),如果设备处于未初始化状态,且设备没有接收到其它的控制信息,设备首先会发送一个 8 Bytes 的中断信息给 USB 主机^[10],如果此中断信息的前 4 字节为 0x00000001 则 USB 主机可以从 Generic RNDIS 设备读 Response to REMOTE_NDIS_INITIALIZE_MSG 报文,若此报文的状态位为 SUCCESS 则 Generic RNDIS 设备进入初始化状态。

在 USB 层上表现为 USB 主机发送一个 SEND_ENCAPSULATED_COMMAND 给设备的控制端点,设备收到后通过中断端点发送一个 RESPONSE_AVAILABLE 给 USB 主机,若状态为 SUCCESS 则 USB 主机从设备控制端点读 GET_ENCAPSULATED_RESPONSE,若状态为 SUCCESS 则设备进入初始化状态,其流程如图 4 所示。

接下来 USB 主机给 Generic RNDIS 设备发送一个设置信息报文 RNDIS_SET_MSG,在其中的 OID(标识字段)字段设为 OID_GEN_CURRENT_PACKET_FILTER 且参数为非 0 则 Generic RNDIS 设备从初始化状

态进入数据初始化状态。Generic RNDIS 设备配置流程如图 5 所示。

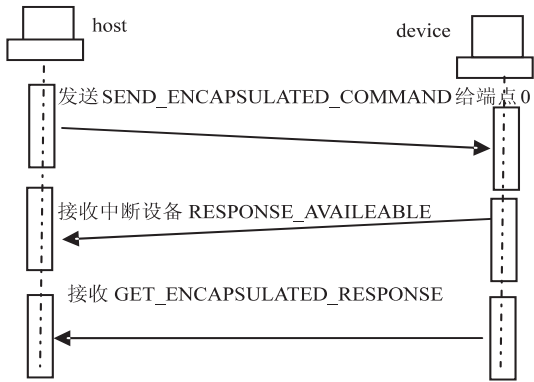


图 4 RNDIS 协议在 USB 层的流程

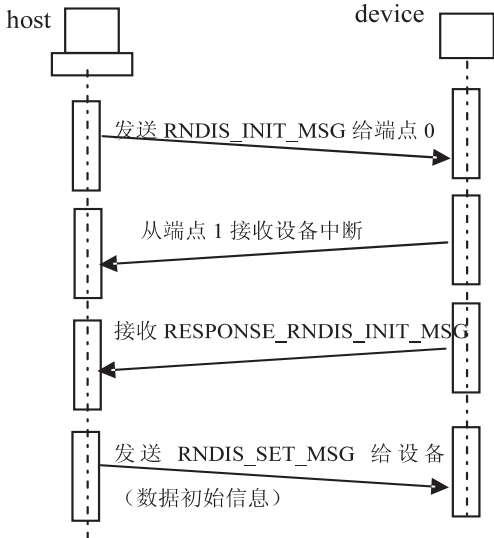


图 5 RNDIS 协议初始流

2.4 Generic RNDIS 设备数据通信过程

Generic RNDIS 设备进入数据初始状态后还需要一系列的配置才能实现网卡的功能,首先 USB 主机它会发送一个包含自己 MAC 和 IP 地址的 UDP 广播^[11],该广播向网络内所有连通的设备发送,而 Generic RNDIS 设备收到该 UDP 广播后会连续发送两个包含自己 MAC 和 IP 地址的 ARP 广播,回复 USB 主机的 UDP 广播,USB 主机收到 RNDIS 设备的 ARP 广播后继续发送 UDP 数据包给 RNDIS 设备以确认它已收到 RNDIS 的 MAC 和 IP 信息,Generic RNDIS 设备也回一个 UDP 确认包,并重复一次这个 UDP 广播交互过程。然后 USB 主机连续发送三个 ARP 广播,Generic RNDIS 设备收到后将发送一个 DHCP 广播,USB 主机收到后发送一个 ARP 确认广播,Generic RNDIS 设备回复一个收到 ARP 广播,这里所提到的 ARP 广播、UDP 广播及 DHCP 广播都是程序按照相应网络协议标准实现的,例如 ARP 广播是以 RFC 826 文档标准中定义的内容来实现。至此经过一系列的交互,USB 主机和 Generic RNDIS 设备都获了彼此的 MAC 和 IP 地址,实现了网卡功

能,接下来就可以发送一个真正的报文了。数据报文是以 UDP 数据包的形式来发送,UDP 数据包的组成是按照 RFC 826 文档标准中的内容来实现,UDP 数据包按标准中的各个字段来填充,并计算检验和等信息。Generic RNDIS 设备数据配置流程如图 6 所示。

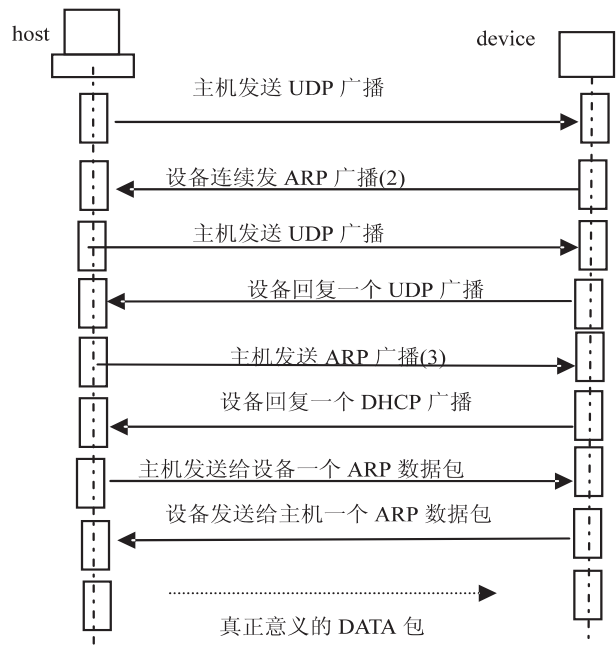


图 6 虚拟网卡配置流程

关于在 Generic RNDIS 设备和主机之间的报文格式可以参考微软的 RNDIS 协议,里面有详细的描述,而 UDP 有关的结构定义则可以参考 UDP 协议及相应的标准文档,这里将不再详细描述。只给出其方法实现。

2.5 实验结果

在 dopod 手机(该手机使用 Windows mobile 5.2 OS 内置了 RNDIS 功能)平台上实现了上述程序^[12],在 dopod 手机上测试了程序的功能实现和性能分析,分别在手机端(RNDIS 设备)和 PC 机(USB 主机)运行客户端程序和虚拟网卡程序,分别传送 100M 字符数据和文件数据,分别存储在手机 SD 卡中,然后将这 100M 数据再修改后再回送给 PC,最后用 BUS HOUND 来捕获 USB 级的通信数据信息,用 Wire Shark 来捕获网络层的通信数据信息,经分析和实验表明该程序可保证正确无误地传递相应字符数据和文件数据。所以,在

主机端,WIN XP 平台已经把 Generic RNDIS 设备识别为一个具有收发功能的以太网网卡,只要设备初始化完成就可以发送接收以太网数据。

3 结束语

只要设备上有对 RNDIS 功能的支持,程序在设备上面实现是可行的,PC 机上的程序会把设备虚拟成一个以太网网卡来传输数据,程序实现了微软 ActiveSync 工具的同步功能,微软的 ActiveSync 是基于 TCP 协议的,本程序是基于更为简单的 UDP 协议,而 UDP 协议不保证可靠传输,程序实现时要加上回复确认机制,以实现 TCP 级别的可靠传输。因此本程序比 ActiveSync 更为简单易行,当前智能手机市场的激烈竞争和 3G 网络的发展,微软也把原来的 Windows mobile 升级为 Windows Phone 以对抗 Android 手机系统,随着 Windows Phone 手机市场份额的增加,基于 USB 接口的 RNDIS 数据通信也将有更广泛的应用。

参考文献:

- [1] Microsoft Corporation. Remote NDIS Specification Rev1.1 [S/OL]. 2002. <http://www.microsoft.com>.
- [2] Microsoft Corporation. NDIS Specification Rev1.0 [S/OL]. 1995. <http://www.microsoft.com>.
- [3] Microsoft Corporation. Remote NDIS (RNDIS) and Windows [S/OL]. 2009. <http://msdn.microsoft.com>.
- [4] 虞科华. 基于 RNDIS 规范的 USB 网络设备设计[J]. 测控技术, 2007, 26(5): 111-113.
- [5] Johannes Libusb-0.1 API Documentation [S/OL]. 2010. <http://www.libusb.org/>.
- [6] 刘 静, 耿国华. 基于 USB2.0 的高速大容量数据采集存储系统[J]. 计算机技术与发展, 2011, 21(2): 143-146.
- [7] USB implementer's forum. USB specification rev2.0 [S]. CompaqIntelMicrosoftNEC, 1998.
- [8] 王云飞. USB 系统研究[D]. 北京: 清华大学, 2001.
- [9] Axelson J. USB 开发大全[M]. 李鸿鹏, 郑瑞霞, 陈香凝译. 北京: 人民邮电出版社, 2011.
- [10] 马 伟. 计算机 USB 系统原理及其主从机设计[M]. 北京: 北京航空航天大学出版社, 2004: 39-59.
- [11] UDP 协议[S]. IETF RFC 768, 1980.
- [12] 谭浩强. C 程序设计[M]. 北京: 清华大学出版社, 1997.

(上接第 84 页)

- [J]. 计算机技术与发展, 2007, 17(7): 244-246.
- [10] MSDN LIBRARY. . NET Framework Regular Expressions [EB/OL]. 2006-06-09. [http://msdn.microsoft.com/en-us/library/hs600312\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/hs600312(VS.80).aspx).

- [11] Friedl J E F. 精通正则表达式[M]. 余 晟译. 北京: 电子工业出版社, 2009.
- [12] 沙 金. 精通正则表达式 - 基于 .NET/ASP/PHP/JSP/JavaScript[M]. 北京: 人民邮电出版社, 2008.

基于RNDIS协议的虚拟网卡程序实现

作者: [高东飞, 张志鸿](#)
作者单位: [郑州大学 信息工程学院, 河南 郑州 450000](#)
刊名: [计算机技术与发展](#)
英文刊名: [Computer Technology and Development](#)
年, 卷(期): 2013(2)

本文链接: http://d.g.wanfangdata.com.cn/Periodical_wjfz201302023.aspx