·网络与通信· 文章编号：1000—3428(2003)21—0125—03 文献标识码：A 中图分类号：TP316

在蓝牙无线网络中虚拟网卡的实现

李 萌，杨卫华，张 林
（清华大学电子工程系，北京100084）

摘 要：介绍了基于Windows 2000系统且符合NDIS规范的虚拟网卡驱动程序的主体框架，并给出了其在蓝牙无线网络中的一个应用实例。

关键词：虚拟网卡；蓝牙；驱动程序

Implementation of Virtual Network Miniport Driver in the Bluetooth Network

LI Meng, YANG Weihua, ZHANG Lin
(Department of Electronic Engineering,Tsinghua University, Beijing 100084)

【Abstract】 This article introduces the framework of a virtual network miniport driver embedded in the Windows 2000 NDIS infrastructure.An application of it in the context of Bluetooth network is also presented.

【Key words】 Virtual network miniport driver; Bluetooth; Driver

在当前主流的操作系统由Windows 95向Windows 98，Windows 2000过渡的情况下，驱动程序的开发也由原先Windows 95下的VXD(虚拟设备驱动程序)发展为Windows 98,Windows 2000下结构更合理，效率更高的WDM(Windows驱动程序模型)。NDIS(Network Driver Interface Specification,网络驱动接口标准)是基于WDM驱动程序开发的一个子集，它将网络驱动分为Miniport Driver，Intermedia Driver和Protocol Driver 3层，并在各层之间提供一致的接口与标准，使得网络驱动的开发更加便捷，思路更加清晰。

NDIS的3层结构为每一层定义了不同的功能并面向不同的对象。Miniport Driver层直接管理物理网卡，并对上层提供一致的接口，以屏蔽不同物理网卡实现相同功能时的技术细节；Protocol Driver层向上实现TDI的接口来对网络的使用者提供服务，向下则对Miniport Driver提供收发数据包的接口；Intermedia Driver层间于Miniport Driver和Protocol Driver之间，进行例如数据包过滤、转换等工作。若单单从实现虚拟网卡所要求的截获系统所发的数据包这一要求来看，3层的Driver都可以实现，但考虑到上层协议的绑定，以及Intermedia Driver必须要附着于已有的Miniport Driver驱动这一要求，使用Miniport Driver的框架来开发虚拟网卡驱动就成了必然的选择。

虚拟网卡驱动程序对于操作系统而言就像一块真正的物理网卡，操作系统可以对其进行所有真正物理网卡可以进行的控制，如收发数据包、禁用、启用、挂起、查询状态和绑定TCP/IP、IPX协议等。将虚拟网卡与蓝牙无线技术结合起来，操作系统可以在其上正常使用HTTP，FTP，Telnet等Internet服务和网络邻居等局域网应用，并可实现NAT等高级应用。本文论述了在Windows 2000操作系统中，使用Microsoft提供的2000 DDK（Device Drivers Kit,设备驱动程序开发包）开发符合NDIS规范的虚拟网卡驱动程序的方法以及如何与蓝牙无线技术相结合实现无线网络。

1 虚拟网卡驱动程序架构分析

为了能对上层呈现为一块物理网卡，虚拟网卡驱动程序（Virtual Network Miniport Driver，VNMD）继承了一般Miniport Driver的结构，能响应NDIS对Miniport Driver提供的各个例程的调用。在本应该由具体硬件实现的收发包等功能上，VNMD通过IRP分发例程，与处于ring 3层的Win32用户态程序进行通信，并由Win32程序控制蓝牙无线设备进行数据包的收发（图1）。

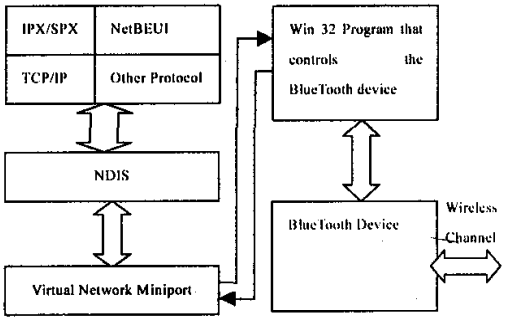

图1 数据包流程

在实际工作中，我们针对CSR(Cambridge Sillicon Radio)公司的BlueCore 01芯片，利用其开发平台CASIRA开发了能实现蓝牙设备间数据传输的应用程序，然后利用VNMD把蓝牙设备封装成对上层应用透明的无线网卡（图2）。

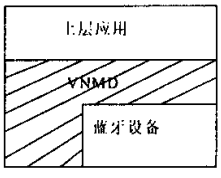

图2 系统功能框图

2 VNMD主要例程及分析

2.1 驱动程序初始化

作者简介：李 萌（1979—），男，硕士，主要研究方向：Ad hoc网络；杨卫华，硕士；张 林，博士、讲师

收稿日期：2002-09-18 **E-mail**：reny@mail.tsinghua.edu.cn

DriverEntry()是驱动程序的初始化入口点，当驱动程序被装入时，内核调用DriverEntry()例程。

部分程序及说明如下：

```
NDIS_MINIPORT_CHARACTERISTICS
NE2000Char; //VNMD对象
NDIS_STATUS Status;//函数返回状态
NdisMInitializeWrapper(&NdisWrapperHandle,…);
//获得NdisWrapperHandler
//设置各回调(callback)例程入口
NE2000Char.QueryInformationHandler=Ne2000QueryInformation;
//读取参数例程入口
NE2000Char.SendHandler = Ne2000Send;
//发送数据包例程入口
NE2000Char.SetInformationHandler = Ne2000SetInformation;
//设置参数例程入口
NE2000Char.TransferDataHandler = Ne2000TransferData;
//接收数据包例程入口
…//其他一些必备例程入口
Status = NdisMRegisterMiniport(…);
//向内核注册VNMD所用到的回调例程
//为与Win32程序进行通信进行的准备工作如下
NdisInitUnicodeString(&DeviceName,L"\\Device\\VNMD");
//初始化设备名
NdisInitUnicodeString(&SymbolicName,L"\\DosDevices\\VNMDs
");//初始化符号链接名
MajorFunction[IRP_MJ_CREATE]=BTCreate;
//与Win32 CreateFile()对应的回调例程
MajorFunction[IRP_MJ_CLOSE]=BTClose;
//与Win32 CloseHandle()对应的回调例程
MajorFunction[IRP_MJ_READ]=BTRead;
//与Win32 ReadFile()对应的回调例程
MajorFunction[IRP_MJ_WRITE]=BTWrite;
//与Win32 WriteFile()对应的回调例程
Status = NdisMRegisterDevice(…);
//在系统的设备栈中注册设备名和相应的符号链接名以及处理
//IRP的各个回调例程
DeviceObject->Flags |= DO_BUFFERED_IO;
//使得AssociatedIrp.SystemBuffer可读写
```

如果以上步骤皆返回NDIS_STATUS_SUCCESS，则DriverEntry()返回NDIS_STATUS_SUCCESS以标识VNMD被正确初始化，其后内核将调用在DriverEntry()中注册的InitializeHandle例程来进行VNMD的进一步初始化；若以上步骤不能正确完成，则调用NdisTerminateWrapper()以销毁内核分发的NdisWrapperHandle，DriverEntry()返回NDIS_STATUS_UNSUCCESSFUL以标识VNMD未能被正确初始化。

2.2 数据包的发送

Miniport Driver可以选择注册MiniportSend或者MiniportSendPackets作为发送数据包的回调例程，前者一次只发送一个包，后者一次可以发送多个包，当两者都注册时，NDIS只调用MiniportSendPackets来发送数据包。VNMD中注册了MiniportSend作为发送数据包的回调例程。VNMD中的MiniportSend回调例程只完成简单的工作，即将收到的数据包迁移到发送数据包链表的末端，等待Win32用户态程序的处理，并返回NDIS_STATUS_PENDING，以提示NDIS上层该数据包尚未处理完毕，VNMD仍然保有该数据包的使用权。Win32用户态程序采用轮询机制，间隔固定的时间就执行ReadFile()函数，内核继而发出I/O请求包(IRP)由与IRP_MJ_READ相对应的BTRead()回调例程来执行。BTRead()读取数据包中的数据写入到一整块Buffer中，返回给Win32用户态程序。而后Win32用户态程序就可以将数据Buffer通过蓝牙无线设备发送出去，对等端接收后，做相应的接收处理。

2.3 数据包的接收

本地蓝牙无线设备接收到对等端发送过来的数据Buffer后，Win32用户态程序执行WriteFile()函数，内核发出IRP由与IRP_MJ_WRITE相对应的BTWrite()回调例程来执行。BTWrite()从IRP中读出数据包Buffer值，将其包装成一个符合NDIS规范的数据包，再将其送到NDIS上层。对于无连接的（Connectionless）Miniport Driver来说，往上层送数据包有两种方式：一是使用NdisMIndicateReceivePackets()，将收到的若干个数据包完整地传递给NDIS上层；一是使用NdisMEthIndicateReceive()，将收到的一个（且只能为一个）数据包的包头和N Bytes的LookAhead传送给NDIS上层（N取数据包包长和Max_LookAhead中的最小值），对此包感兴趣的上层驱动会继而调用MiniportTransferData()回调函数来获得该数据包的完整数据。实际使用中发现，对于NDIS3.0版本，使用NdisMIndicateReceivePackets()对上层送数据包与某些网络协议不兼容，使用EtherPeek等数据包截获工具可以发现，NDIS上层对Arp的数据包可以作出响应，但是对ping的数据包则毫无反应。改用NdisMEthIndicateReceive()来向上层传送数据包时，则VNMD工作正常。

2.4 系统参数设置与读取

MiniportSetInformation()&QueryInformation()是NDIS用来设置和读取Miniport Adapter参数的回调例程。VNMD正确初始化后，NDIS会调用MiniportQueryInformation读取一系列的Miniport Adapter参数值供上层驱动与协议使用，如：Mac值，最大LookAhead值，最大包长等；而后NDIS会根据上层驱动的需要，调用MiniportSetInformation设置相应的参数值，如多播地址(Multicast Address),最大LookAhead值等。

3 调试

驱动程序与Win32用户态程序很大一点的不同就在于驱动程序没有内核的内存保护，任何的程序错误都可能导致系统的崩溃，且驱动程序的调试也不如Win32用户态程序方便。VNMD的开发过程中主要使用Chris Cant所著"Writing Windows WDM Device Drivers"一书中提供的DebugPrint工具输出调试信息以进行调试。基本完成驱动程序后，使用WildPackets公司提供的EtherPeek NX工具可以实时分析VNMD发出和接收的数据包，为进一步完善驱动程序提供了极大的方便。

程序基本完成时，调试时曾发现系统欲发送的数据包长达几MB，而实际系统中的数据包长度一般只有几百到几千字节。研究后发现问题是由于系统发包与Win32用户态程序读包的时间间隔超过了ndis预定的2s，系统reset了，导致数据包被丢弃，是实际读到的无效数据造成的。通过在NdisMSetAttributesEx()里延长系统CheckForHang的间隔，可以解决问题。而后实际读到的有效包长一般在几百个字节。

4 结语

VNMD（虚拟网卡驱动程序）可以看成是Miniport Driver的继承类，它具有一般Miniport Driver的结构和一些相应的功能函数，但又有其独特的实现方法。如必须调用NdisMRegisterDevice()来注册设备名和相应的符号链接名，

并提供相应的回调例程来处理IRP，使得其能与Win32用户态程序进行通信。在收发数据包的机制上，也不同于一般的Miniport Driver：发送数据包时采用Win32用户态程序polling的方式，接收数据包时不可使用一次传多个包的NdisMIndicateReceivePackets()例程，而必须使用NdisMEthIndicateReceive()和MiniportTransferData()例程。

在具体应用场景下，考虑到蓝牙无线链路的特殊性，我们也对VNMD进行了算法和数据结构上的优化，以增加链路的带宽。比如针对蓝牙无线设备通信时握手时间长、开销大，则考虑Win32用户态程序使用polling机制发数据包时，可以一次读取多个数据包合并成一个大数据包后再利用一次单向信道发送出去，接收端收到大数据包后再进行解包的工作，还原原先的多个数据包并依次送到VNMD. 在实际应用中，每次蓝牙无线设备握手时间大概为0.1s，传送一个1kB的数据包，需要1*8/(721kbps)=0.01s的时间，系统效率只有10%，优化后，在进行FTP等大数据量不对称传输时，每次可以平均发送10kB的数据，系统效率可以达到50%，提高了4倍。同时通过在NdisMSetAttributesEx()中延长系统CheckForHang的时间间隔，可以降低上层协议重发包的概率，避免重传包的系统带宽的消耗。

在实际应用中，VNMD结合蓝牙无线技术实现的蓝牙无线网络可以完全无缝地与操作系统结合起来，在其上可进行所有网络操作，如ping，telnet，网络邻居，FTP, HTTP等。速率与一般modem的速率相近。在蓝牙标准的下一个版本达到10Mbps速率后，该蓝牙无线网络则可以基本满足宽带应用的需求。

除了蓝牙无线网络，VNMD还有其他更广阔的用途，如结合其他数据物理链路实现虚拟网络，并可以进行数据包分析、网络协议原理研究等。

参考文献

1 Cant C. Writing Windows WDM Device Drivers. 北京：机械工业出版社，2000

2 Miller B A, Bisdikian C. Bluetooth Revealed. 北京：人民邮电出版社，2000

3 Microsoft Co. Mirosort Windows 2000 Driver Design Guide. 北京：北京大学出版社，2000

4 http://www.csr.com.

（上接第42页）

(2) 作为医学研究和教学工具，系统使用了面向术式的手术模拟交互。让医生在虚拟环境中容易地进行手术规划和手术过程模拟，比较不同的手术方案和结果，以帮助制定出最佳手术方案和提高手术的安全性。

(3) 根据三维重构或手术模拟所得的几何描述，通过模型输出，向力学分析和计算机辅助制造系统设计与制造人体的假肢提供数据。

参考文献

1 Richard R A, Dennis H P, Camp J J. Computer-aided Surgery Planning and Rehearsal at Mayo Clinic. IEEE Computer, 1996, 29 :39-47

2 Handels H, Ehrhardt J, Plotz W.Virtual Planning of Hip Operations and Individual Adaption of Endoprostheses in Orthopaedic Surgery. International Journal of Medical Informatics,2000, 58-59: 21-28

3 Blackwell M, Simon D, Rao S. Design and Evaluation of 3-D Preoperative Planning Software: Application to Acetabular Implant Alignment .Carnegie Mellon University, The Robotics Institute, 1996

4 Montani C, Scateni R, Scopigno R. Decreasing Isosurface Complexity Via Discrete Fitting. Computer Aided Geometric Design,2000,17: 207

（上接第124页）

体类别的分类性能，而宏平均反映的是对所有类别的分类性能，分类效果的优化是根据具体应用背景进行的。本系统中认为各个类别的重要性相同，系统追求的是在各个类别的微平均值相近的情况下使宏平均值尽量高。

3 实验与结论

实验中利用《物理学科分类法》中的10个类，包括总论、基本粒子物理学与场、核物理学、原子和分子物理学、唯象论的经典领域等，使用物理文摘约25 000篇，每类约2 500篇，构建数据集，通过10-折交叉评价方法训练和测试分类器，分类器的宏平均召回率为77.2%，宏平均准确率为77.2%。改进后抓取器与改进前抓取器进行对比测试，测试环境为Windows 2000，奔腾Ⅲ CPU，主频800MHz，内存256MB，通过百Mpbs局域网用户共享2 Mbps出口带宽，给予相同种子站点，每个站点最多抓取10 MB，只抓取网站的最上面两层，小于1 000个页面，数据如下：

测试对象	运行时间	种子站点数	抓取网站数目	物理类网站数目	准确率
抓取器	8小时	250	522	53	10.1%
改进后抓取器	8小时	250	134	73	54.4%

从测试结果上看，基于自动分类的网页抓取器提高了抓取用户感兴趣网页的效率，抓取的网站数目是未改进前的25%，抓取网站的数目大大减少。抓取的准确程度大大增加，由10.1%增加到54.4%。由于抓取器抓回的网页已经给出了类号，也提高了对网页后续处理的效率。影响实验的主要因素有：种子站点的选取，网络的繁忙情况，分类器的分类准确程度。较好的种子站点是本类别中权威程度较高的站点，其中包含指向本类别网站的URL较多，这样准确度会提高。相反，则运行时间越长准确度越低，抓取回的无用网页越多。分类器由于受到学习样本空间和真正使用空间的差异及分类算法等的影响，测试的准确率往往高于实际使用的准确率，而那些分错的结果又会带入下一次抓取，结果导致运行时间越长，准确率越低，所以系统运行一段时间后要停下来，由人对结果进行评价，进行数据整理，提供高质量的种子，进行下一次的运行。

参考文献

1 Yang Yiming. An Evaluation of Statistical Approaches to Text Categorization. Journal of Information Retrieval, 1999, 1(1/2): 67-88

2 卜东波. 聚类/分类理论研究及其在文本挖掘中的应用[博士学位论文]. 北京：中科院计算所，2000

3 刘　斌，黄铁军，程　军等.一种新的基于统计的自动文本分类方法. 中文信息学报，2002，(6)

在蓝牙无线网络中虚拟网卡的实现

作者： 李萌， 杨卫华， 张林
作者单位： 清华大学电子工程系,北京,100084
刊名： 计算机工程 ISTIC PKU
英文刊名： COMPUTER ENGINEERING
年，卷(期)： 2003,29(21)

参考文献(4条)

1. Cant C Writing Windows WDM Device Drivers 2000

2. 查看详情

3. Microsoft Co Mirosort Windows 2000 Driver Design Guide 2000

4. Miller B A;Bisdikian C Bluetooth Revealed 2000