

计算机网卡状态快速诊断

周 鸿¹,董张卓²,唐 明²

(1.西安科技大学,陕西 西安 710054;2. 陕西银河电力自动化股份有限公司,陕西 西安 710075)

摘要: 计算机网络驱动程序接口规范 NDIS(Network Driver Interface Standard)协议能实现快速捕获计算机网卡状态。介绍了 NDIS 网络接口规范、架构,给出了驱动程序模型及程序执行泳道图,编写网络状态诊断方法的 NDIS 协议驱动程序及应用程序。测试结果说明用 NDIS 驱动程序方式对网卡工作状态的诊断效率高,其诊断时间为毫秒级,可满足电力调度主站系统实时快速的要求。

关键词: NDIS; 协议驱动程序; 网卡状态; 快速诊断

中图分类号: TM 734

文献标识码: A

文章编号: 1006-6047(2005)01-0075-04

0 引言

目前的调度自动化系统普遍采用分布式的双机双网或单网计算机系统^[1]。调度主站系统软件中,计算机之间的通信采用 TCP 或 UDP 协议,用套接字(Socket)建立通信,但是用套接字建立通信链路,不能及时得到网络或网卡的通信状态。计算机网络状态的诊断一般用网间控制报文协议 ICMP(Internet Control Message Protocol)报文对计算机网络通信状态进行探测和诊断,根据 ICMP 协议的特点,网络管理诊断软件运行时,只能采用轮询方式,因此,按此原理设计的网络诊断软件,运行时将占用较多的计算机 CPU 资源,且实时性较差^[2]。

实现对调度主站网络快速诊断的关键点之一,是能够快速准确对网卡的状态进行诊断。应用网络驱动程序接口规范 NDIS(Network Driver Interface Standard)能够解决问题^[3]。NDIS 通过对其库函数的调用,能够对计算机网卡进行高效的管理,从而实现网卡状态的诊断,提高系统的效率。因此,采用编写 NDIS 驱动程序获知网卡的状态。

本文首先介绍 NDIS 网络接口规范的架构,在此基础上采用统一建模语言 UML(Unified Modeling Language)技术建立网卡驱动程序的模型^[4],并编制了网卡驱动程序和诊断验证程序。

1 NDIS 网络接口规范

NDIS 定义网卡驱动程序与上层协议驱动程序之间的通信接口规范,通过 NDIS 规范协议层和物理层的接口,使上层的协议驱动程序可以和底层任何类型的网卡通信。实现传输驱动程序和网卡驱动程序之间的相互通信,及网卡驱动程序在逻辑上与传输驱动程序之间的接口。按照 NDIS 接口标准,任何与 NDIS 兼容的传输驱动程序都能与 NDIS 兼容的网络

适配器驱动程序进行信息交换。NDIS 架构如图 1 所示,分 3 个层次:网卡驱动(微端口驱动)、中间驱动、协议驱动^①。

微端口驱动程序是网卡与上层驱动程序间通信的接口,可以管理 1 个网络接口卡 NIC(Network Interface Card),包括通过 NIC 发送和接收数据,并且与高层驱动程序相接。

中间驱动程序位于微端口驱动程序和协议驱动程序之间,它向上提供小端口(miniport)函数集,向下提供协议(protocol)函数集。因此,对于上层驱动程序而言,它是小端口驱动程序;对于底层驱动程序,它是协议驱动程序。能够实现不同包介质的转换,过滤包及多网卡时的包传送。

协议驱动程序执行具体的网络协议,例如网间数据包交换 IPX(Internet Packet eXchange)/顺序包交换 SPX(Sequences Packet eXchange),TCP/IP。能够使用操作系统提供的功能,读取、设置、发送包给微端口 NIC 驱动程序和中间驱动程序,还为应用程序访问低层驱动程序提供接口。

从 NDIS 驱动程序的结构图中看出,NDIS 作为网卡与网卡驱动程序及与上层驱动程序之间的接口规范,可以通过 I/O 管理程序或事件共享内存等方式实现驱动程序与应用层之间的通信。

2 网卡状态诊断方法

根据 NDIS 的架构,能通过编制协议驱动程序,实现网卡状态的快速诊断。在协议驱动程序层捕获

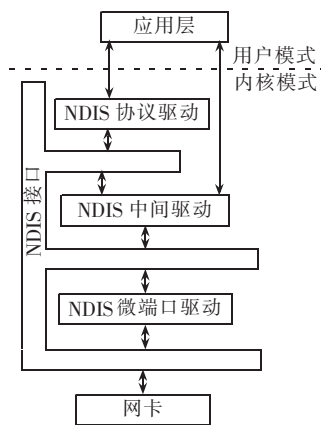


图 1 NDIS 结构

Fig.1 The structure of NDIS

网卡的状态,在应用层得到网卡状态。程序分为协议驱动程序和应用层程序。

NDIS 协议驱动程序能够在网卡状态变化时及时获知网卡状态变化信息,并将这一信息快速传送给应用程序作相应的处理。网卡驱动程序运行时自行检测网线的连接状态,如果发生改变,例如网线插拔、网卡的禁用与启用,网卡设备驱动程序能通知所有的上层 NDIS 协议驱动程序。利用协议驱动程序接收网卡状态改变这一特点,开发相应的 NDIS 协议驱动程序,并将状态改变信息传送给应用层实现网卡状态的诊断。

应用程序与驱动程序之间通信方法^[5,6]是:在应用程序里设置 1 个 Event,启动 1 个线程等待事件发生,然后用 DriverIoControl 把这个 Event 传给驱动程序,驱动程序用 ObReferenceObjectByHandle 将这个 Event 转成内核的 Event。当状态发生改变时,用 KeSetEvent 发信号通知应用程序,应用程序通过 DriverIoControl 从驱动程序读取数据。

信息的传送过程为协议驱动程序接收下层驱动或 NDIS 发送的网卡状态改变信息,并利用事件通知应用程序,应用程序在接到通知后通过 I/O 请求包 IRP(I/O Request Packet)获得相应的网卡状态改变信息,如图 2 所示。

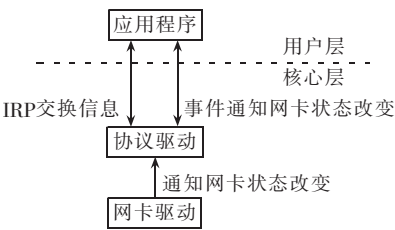


图 2 信息传送过程

Fig.2 The process of information transfer

3 驱动程序和应用测试程序设计

3.1 网卡状态监测程序架构

程序由应用程序 CommApp 和驱动程序 Packet 两部分组成,应用程序与驱动程序通过 DeviceIoControl 通信。驱动程序通过接口 ProtocolStatus 应用 NDIS 动态链接库的相关函数。程序构件图如图 3 所示。

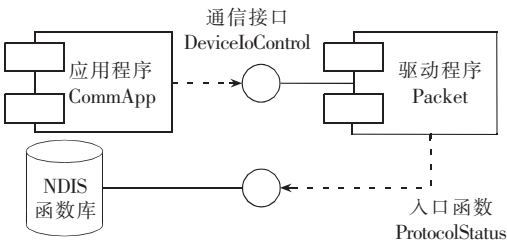


图 3 程序构件图

Fig.3 The program components

3.2 网卡状态监视程序设计

3.2.1 协议 DriverEntry 及其初始化^[7]

驱动程序先初始化入口,且入口点必须被命名

为 DriverEntry。通过执行 NdisRegisterProtocol 注册协议驱动程序入口函数指针和协议名称,入口函数为 ProtocolStatus(),接受句柄状态改变。然后,初始化驱动程序的 dispatch 例程,在驱动程序中能处理 I/O 请求。利用其中的 MajorFunction[IRP_MJ_DEVICE_CONTROL]例程进行用户层与核心层的数据交换。

3.2.2 应用程序向协议驱动程序传送事件句柄

当驱动程序在初始化 dispatch 例程之后,应用程序具备创建事件条件,并启动线程,等待事件,通过 DeviceIoControl 将创建后的事件句柄传递给协议驱动程序,然后,在协议驱动程序中利用 ObReferenceObjectByHandle 获得相应事件句柄的指针,再通过 KeSetEvent 对事件指针进行置位。

3.2.3 协议驱动程序接收下层发送状态改变通知

网卡连线状态若发生改变,即拔去了网线或接上了网线,低层驱动程序就会调用 NdisMIndicateStatus 通知所有上层 NDIS 协议驱动程序。这时,回调函数 ProtocolStatus 被调用,同时,传入指向发生状态改变适配器的指针和状态标识码。其中,利用适配器指针可以得到适配器的系统 service name,状态标识码 NDIS_STATUS_MEDIA_DISCONNECT 指示网线被拔掉,状态标识码 NDIS_STATUS_MEDIA_CONNECT 指示网线被接上,状态标识码 NDIS_STATUS_LINK_SPEED_CHANGE 指示网卡速度改变(10/100 自适应网卡才有)。这样,就可以在 ProtocolStatus 将相关适配器信息和状态信息记录下来。

3.2.4 禁用、启用网卡的处理

当人为禁用、重新启用网卡时,协议驱动程序中的回调函数 ProtocolBindAdapter 和 ProtocolUnbindAdapter 会被调用,同时会传入设备对象和扩展信息。

3.2.5 协议驱动程序利用事件通知应用程序状态改变

当 3.2.3 或 3.2.4 中将相关适配器信息和状态信息记录之后,通过 KeSetEvent 对事件指针进行置位以及通知上层应用程序有状态改变发生。这时应用程序就可在线程中接收到这个事件。

3.2.6 应用程序读取状态改变信息的处理

应用程序读取状态改变信息,并作相应的处理,应用程序中线程 WaitForSingleObject 等到事件后,利用 DeviceIoControl 通过自定义 METHOD_BUFFERED 类型的 CTL_CODE 向协议驱动程序发送 IRP,协议驱动程序将在 3.2.3 或 3.2.4 中相关的适配器信息和状态信息返回给应用程序,通过这些信息,可以进行网卡状态改变的显示、报警等。程序执行的泳道见图 4。

3.3 涉及函数说明^①

3.3.1 应用程序

应用程序函数如表 1 所示。

3.3.2 驱动程序函数

驱动程序函数如表 2 所示。

① Windows 2000 DDK 文档, Microsoft Corporation, 2000, 6.

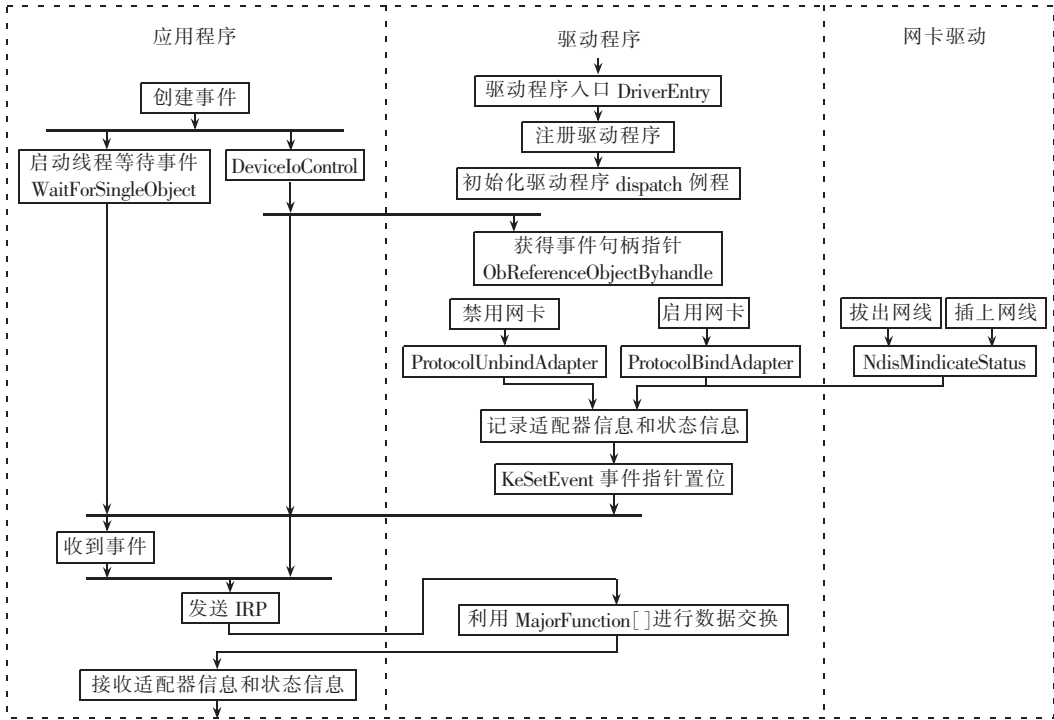


图 4 程序泳道图

Fig.4 The program swimlanes

表 1 应用程序函数

函 数	说 明
DeviceIoControl	初始化驱动程序范围的数据结构和资源
WaitForSingleObject	调用者线程变为等待状态直到事件发生

表 2 驱动程序函数

函 数	说 明
DriverEntry	由操作系统调用来激活和初始化微端口驱动程序
dispatch	驱动程序派遣例程,解释怎样使驱动程序收到相应 I/O 功能代码
ObReferenceObjectByHandle	获得事件句柄指针
PacketBindAdapter	网卡初次进入启用状态时,被 NDIS 调用
PacketUnbindAdapter	网卡初次进入禁用状态时,被 NDIS 调用
NdisCloseAdapter	拆除对低层 NIC 驱动程序的绑定并释放资源
KeSetEvent	对事件指针置位,通知应用程序事件发生
NdisWaitEvent	引起调用者等待到指定事件被指示或指定时间间隔结束时止

表 3 测试结果

ms				
编号	启用网卡	禁用网卡	拔网线	插网线
1	0.024	0.024	0.001	4.752
2	0.032	0.023	0.001	3.450
3	0.033	0.023	0.001	5.753
4	0.032	0.026	0.001	2.749
5	0.016	0.024	0.001	3.750
6	0.032	0.023	0.001	2.749
7	0.005	0.024	0.0009	2.749
8	0.016	0.025	0.001	2.749
9	0.033	0.025	0.001	5.753
10	0.005	0.022	0.001	3.750
平均	0.023	0.024	0.001	3.821

4 实验结果

根据以上设计,用 VC++ 6.0 编制了驱动程序和验证应用程序。采用禁用、启用网卡,拔掉、插上网线方式进行测试,测试结果如表 3 所示。

从表中可得:禁用网卡、启用网卡的平均响应时间约为 0.02 ms,拔掉网线的平均响应时间约为 0.001 ms,插上网线的平均响应时间约为 3.82 ms。由此可知,网

卡状态改变的响应时间均为毫秒级,完全能够满足网络的监控要求。

5 结论

由于 NDIS 规范向驱动程序提供与操作系统进行通信的接口,为通信应用提供了很好的抽象功能,应用程序能使用它进行网络通信和网络管理,使得网络驱动程序和操作系统实现隔离分离,方便了网络驱动程序的编写。通过编写 NDIS 协议驱动程序诊断网卡的工作状态,能够满足电力调度主站系统对网卡工作状态诊断的要求,其诊断时间为毫秒级。

参考文献:

[1] 赵晓东.国内开发的几种电网调度自动化系统的比较[J].供用电,2003,(8):10-16.

ZHAO Xiao-dong. Comparison of several domestic-developed dispatching automation systems[J].Distribution and

- Utilization, 2003, (8):10-16.
- [2] STEVENS W R. TCP/IP 详解 卷 1: 协议[M]. 范建华, 胥光辉, 张 涛, 等译. 北京: 机械工业出版社, 2000.
- STEVENS W R. TCP/IP illustrated volume 1: The protocols [M]. Translated by FAN Jian-hua, XU Guang-hui, ZHANG Tao, *et al.* Beijing: China Machine Press, 2000.
- [3] 李晓莺, 曾启铭. NDIS 网络驱动程序的研究与实现[J]. 计算机应用, 2002, (4): 60-61.
- LI Xiao-ying, ZENG Qi-ming. The research and realise of NDIS-driver[J]. **Computer Applications**, 2002, (4): 60-61.
- [4] BOOCH G, RUMBAUGH J, JACOBSON I. UML 用户指南[M]. 邵维忠, 麻志毅, 张文娟, 等译. 北京: 机械工业出版社, 2001.
- BOOCH G, RUMBAUGH J, JACOBSON I. The unified modeling language user guide [M]. Translated by SHAO Wei-zhong, MA Zhi-yi, ZHANG Wen-juan, *et al.* Beijing: China Machine Press, 2001.
- [5] 翟洪涛. 驱动程序超级宝典 [EB/OL]. <http://www.vchelp.net>, 2004-02-08.
- ZHAI Hong-tao. Treasured book for WIN 2000 driver [EB/OL]. <http://www.vchelp.net>, 2004-02-08.
- [6] 翟洪涛. WIN2000 驱动程序设计 [EB/OL]. <http://www.vchelp.net>, 2004-01-13.
- ZHAI Hong-tao. Design of WIN2000 driver [EB/OL]. <http://www.vchelp.net>, 2004-01-13.
- [7] BAKER A, LOZANO J. Windows 2000 设备驱动程序设计指南 [M]. 施 诺译. 第 2 版. 北京: 机械工业出版社, 2001.
- BAKER A, LOZANO J. The Windows 2000 device driver book [M]. Translated by SHI Nuo. 2nd ed. Beijing: China Machine Press, 2001.

(责任编辑: 汪仪珍)

作者简介:

周 鸿(1977-), 女, 陕西白河人, 硕士研究生, 主要研究方向为电力调度自动化(E-mail: zzhou_hong@163.net);

董张卓(1962-), 男, 山西河曲人, 高级工程师, 博士, 主要研究方向为电力自动化技术和电力系统分析技术;

唐 明(1972-), 男, 浙江舟山人, 工程师, 主要从事电力调度自动化的研发工作。

Rapid diagnosis of network card status

ZHOU Hong¹, DONG Zhang-zhuo², TANG Ming²

(1. Xi'an University of Science and Technology, Xi'an 710054, China; 2. Shaanxi

Galaxy Electric Power Automation Stock Co., Ltd., Xi'an 710075, China)

Abstract: NDIS (Network Driver Interface Standard) protocol helps to capture the network card status quickly. The interface standard and structure of NDIS are introduced. The driver model and swim-lanes of program are presented, and the NDIS protocol driver and application program for network status diagnosis are given. The test results show that the diagnosis efficiency is high and the diagnosis time is several milliseconds, which satisfies the requirements of SCADA (Supervisory Control And Data Acquisition) system for real-time and rapidity.

Key words: NDIS; protocol driver; network card status; rapid diagnosis