

文章编号: 1001-3806(2011)03-0368-04

基于 SNMP 的光纤放大器远程监控系统设计

陶宁, 姜海明*, 肖峻

(电子科技大学 光电信息学院, 成都 610054)

摘要: 为了管理光纤放大器的各项数据指标, 基于简单网络管理协议(SNMP)设计了一种远程监控系统。该系统管理端采用 SNMP++ 开发包为用户提供了友好的管理界面, 代理端使用 MSP430 单片机和 CS8900A 网卡驱动芯片作为网络通信的硬件支持。建立了光纤放大器的各项数据指标管理信息库文件的私有节点。结果表明, 系统通过与实际光纤放大器各功能电路联调测试, 监控系统上各节点对应的各项数据指标均可在用户管理界面中被管理。这说明该监控系统可以用于对光纤放大器进行远程管理。

关键词: 光纤放大器; 简单网络管理协议; 管理信息库; 嵌入式系统

中图分类号: TN943.5; TP393.07 **文献标识码:** A **doi:** 10.3969/j.issn.1001-3806.2011.03.021

Design of remote monitor system of fiber amplifier based on SNMP

TAO Ning, JIANG Hai-ming, XIAO Jun

(School of Optoelectronic Information, University of Electronic Science and Technology of China, Chengdu 610054, China)

Abstract: A remote monitor system was designed to manage the data of fiber amplifiers based on simple network management protocol (SNMP). On the management side, there was a friendly user management interface made by SNMP++. Meanwhile, MSP430, a single chip microcomputer and CS8900A, a network interface card driver chip were used as supporting hardware on the agent side. Private nodes of management information base file were made, which represented objects of the fiber amplifier needing management. Through the union test with an actual fiber amplifier, corresponding private nodes of this system could be managed in the user management interface. It shows that this remote monitor system can be used to manage a fiber amplifier.

Key words: fiber amplifier; simple network management protocol; management information base; embedded system

引言

光纤放大器作为光纤通信系统中的重要组成部分, 分布应用于系统中的各个位置。如何有效地管理光纤放大器的各项指标, 将对整个光通信系统的应用与研究产生现实意义。

简单网络管理协议 (simple network management protocol, SNMP) 是目前应用最为广泛的网络管理协议之一^[1]。SNMP 最重要的指导思想就是要尽可能地简单, 其实质就是应用简单的读写操作, 使网络管理人员可以控制任何支持这种协议的设备。光纤放大器原本不属于网络设备, 然而由于这个特点, 实现对光纤放大器的各项指标进行统一管理成为可能。

基金项目: 国家自然科学基金资助项目 (60588502; 60607005; 60877033); 四川省科技攻关计划资助项目 (2006ZD010-3); 电子科技大学青年科技基金资助项目 (JX0628)

作者简介: 陶宁 (1986-), 男, 硕士研究生, 主要研究方向为光纤通信系统光纤放大器控制与监测技术。

* 通讯联系人。E-mail: jianghm@uestc.edu.cn

收稿日期: 2010-07-08; 收到修改稿日期: 2010-08-11

1 系统组成

系统组成如图 1 所示。SNMP 管理端是一台具有

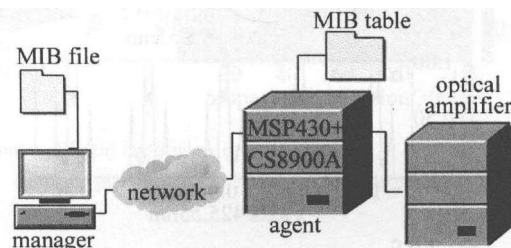


Fig. 1 System constitution

SNMP 应用程序界面的计算机, 用户运用这台计算机进行远程监控操作。代理端平台是具有以太网驱动器的单片机嵌入式系统平台, 它通过网络和管理端通信, 并可实时采集被管理设备——光纤放大器各个被管理对象的数据。另外, 管理端和代理端都建立了关于光纤放大器的私有管理信息库 (management information base, MIB), 用于被管理对象的访问与查找。

2 SNMP 管理端设计

目前有许多可以创建网络管理应用的 SNMP 的应

用程序接口(application program interface, API)。大多数 API 都提供了一个很大的函数库,调用这些函数的程序员需要熟悉 SNMP 内部的工作原理和资源管理机制。SNMP++是一套 C++ 类的集合,它为网络管理应用的开发者提供了 SNMP 服务。SNMP++ 通过提供强大灵活的功能,降低管理和执行的复杂性^[2,3]。由于 SNMP++ 使用 C++ 语言编写,因此充分利用了面向对象的编程技术,SNMP 编程涉及的所有数据结构,全部被封装在相应的类中。所有的底层操作细节,对使用者来说是完全透明的。

作者采用 VC++ 6.0 开发环境,将 SNMP++ 开发包编译生成一个具有完整 SNMP 功能的静态链接库 snmp_pp.lib 文件。通过在建立的管理端应用程序界面工程文件中添加此静态链接库文件,使得所有的 SNMP++ 开发包中的功能函数能像系统调用一样被方便的使用。管理端应用程序界面是基于微软基础类(microsoft foundation classes, MFC)的一个单文档应用程序。程序主要调用 SNMP++ 开发包的功能函数实现 SNMP 协议的基本管理操作,其对应关系见表 1。

Table 1 Management operation & corresponding function

| SNMP management operate | SNMP++ function |
|-------------------------|------------------------------|
| GetRequest | snmp.get(pdu, * target) |
| GetNextRequest | snmp.get_next(pdu, * target) |
| SetRequest | snmp.set(pdu, * target) |
| Trap | set_listen_port(trap_port) |

其中,GetRequest 为管理端请求查询当前被管理对象;GetNextRequest 为管理端请求查询下一个被管理对象;SetRequest 为管理端对当前对象进行设置。snmp.get(pdu, * target), snmp.get_next(pdu, * target) 和 snmp.set(pdu, * target) 3 个对应的函数中,参量 pdu 为包含协议数据单元结构,参量 target 为 1 次 SNMP 通信活动的信息指针。

管理端 Trap 功能是接收代理端主动发送的异常事件信息,是一个被动监听的过程,只需要调用 set_listen_port(trap_port) 监听函数即可,这个函数封装了 Trap 包的接收过程,其中 trap_port 是监听使用的通用端口号参量,通常的传递值是 162。

此外,设计此单文档程序还具有用于读取 MIB 文件的函数 LoadMib(), 查询对象标识符(object identifies, OID)对象节点函数 SearchNode(), 树形控件响应函数 OnItemexpandedTree1() 和 OnSelchangedTree1() 等,用以丰富管理端界面软件的功能,方便网络管理员进行操作。最终实现的 SNMP 管理端界面见图 2。

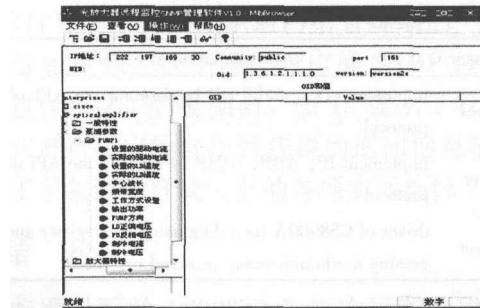


Fig. 2 SNMP software interface of the optical amplifiers

3 SNMP 代理端设计

代理端硬件嵌入式平台是由 MSP430 单片机和网卡驱动芯片 CS8900A 组成。CS8900A 是一款 16 位的低功耗以太网控制芯片。它集成了以太网传输实现时所需要的所有数字电路和模拟电路,还包括丰富的功能和配置选项。CS8900A 提供了 MEMORY, I/O 和 DMA 3 种操作模式,它们各有优缺点。I/O 模式是其默认的工作模式并一直有效,且可以使用 8 位数据总线进行数据读写,即 8 位 I/O 模式,作者就是采用这种 8 位 I/O 模式^[4]。代理端框图如图 3 所示,单片机控制单元电路包括电源复位(power &

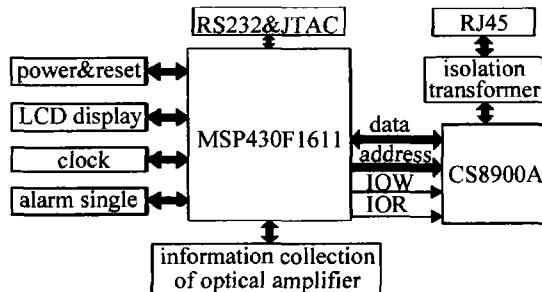


Fig. 3 SNMP software interface of the optical amplifiers on the agent side
reset)、LCD 显示(LCD display)、时钟、告警信号、光放大器模拟量采集,网卡模块电路包括(cs8900A 网卡芯片)、隔离变压器,接口电路包括 RS232, JTAG 和 RJ45 3 种接口。

在 MSP430 和 CS8900A 的连线中,data 表示双向 8 位 I/O 数据总线,address 表示地址总线,用于单片机向网卡读取地址,IOR 表示读操作允许控制,IOW 表示写操作允许控制。

SNMP 协议属于 TCP/IP 协议族中的应用层协议^[5]。针对 TCP/IP 协议族分层体系结构的特点,代理端基于 MSP430 单片机的 SNMP 驱动程序也遵循分层体系的原则,采取逐层实现的方法,将各层协议按照统一的应用程序接口进行连接,从而完成整个协议栈的设计。表 2 中显示了此驱动程序中 TCP/IP 协议族各层对应实现的协议和功能模块框架。

Table 2 Framework of SNMP driver

| | |
|-------------------|-------------------------------------------------------------------------------------------------|
| application layer | implementation of SNMP |
| transport layer | implement TCP and UDP by forming the API of each protocol |
| network layer | implement IP, ARP, ICMP by forming the API of each protocol |
| data-link layer | driver of CS8900A including using the register and processing a ethernet-frame received or sent |

在数据链路层实现了 CS8900A 的硬件驱动包括各种寄存器的读写和以太网帧的收发操作;网络层构建了因特网协议(internet protocol, IP)、地址解析协议(address resolution protocol, ARP)和因特网控制报文协议(internet control message protocol, ICMP)的 API;传输层构建传输控制协议(transport control protocol, TCP)和用户数据包协议(user datagram protocol, UDP)的 API;在应用层最终实现 SNMP。

SNMP 接受来自传输层 UDP 的服务,所以驱动程序的主干是 CS8900A 硬件驱动、IP 协议的 API,UDP 协议的 API 以及 SNMP 功能的实现,其它协议模块作为网络通信测试使用。由于 SNMP 代理端的作用主要是被动的响应管理端请求,所以整个驱动程序可以看作是一个解析处理接收数据帧的过程,如图 4 所示。

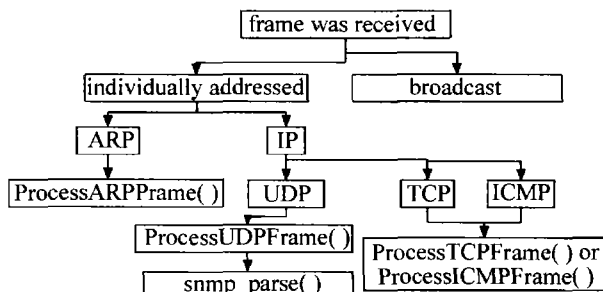


Fig. 4 Demultiplexing of received Frames

处理 UDP 数据帧的函数 ProcessUDPFrame() 包含了 SNMP 数据包的解析处理过程。当接收到 UDP 包,处理 UDP 包头后,再由子函数 snmp_parse() 函数处理 SNMP 包。基本流程见图 5。判断解码标志位是

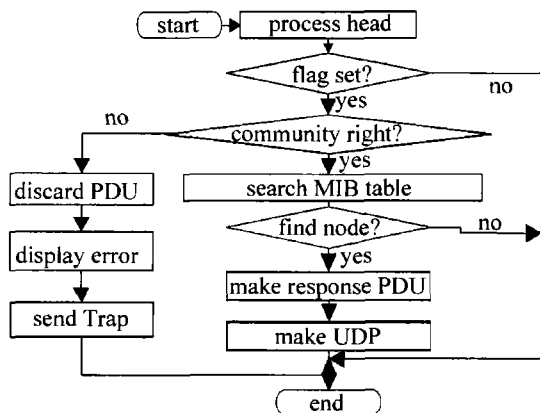


Fig. 5 The flow chart of SNMP packet in agent

否置位,表示解码成功与否,如果解码正确,判断共同

体的是否具有可以进行操作的权限,如果没有权限,则抛弃解析到的协议数据单元(protocol data unit, PDU),在代理端屏幕上显示错误信息,并主动向管理端发送 Trap 消息。如果有操作权限,则查找存贮在代理端内存中的静态 MIB 表,如找到相应节点,则立即执行管理端要求的相应操作,并生成响应报文准备传递给管理端,这样符合 SNMP 协议即时响应的特点。

4 光纤放大器私有节点

MIB 定义了可以通过简单网络管理协议进行访问的管理对象的集合,是被管理对象的虚拟数据库。最初的 RFC1156 定义的 MIB 标准为 MIB-I,接下来的 RFC1213 定义了正式的标准称为 MIB-II,它是 MIB-I 的扩充,增加了一些组和对象^[6-7]。MIB 可以描述为一棵树,各个数据项构成了树的叶结点。然而 MIB-II 中,多数节点都是用于管理常用的网络设备信息,要管理自己特有的设备,则需要在 MIB-II 的私有节点下扩展 MIB 库,建立关于光纤放大器相关内容的私有节点。

如图 6 所示,针对光纤放大器中需要被管理的对

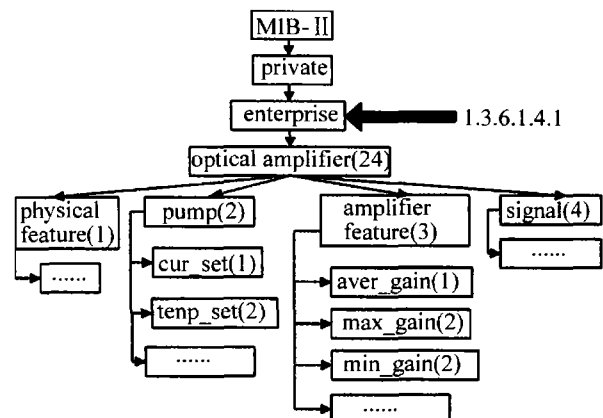


Fig. 6 The private nodes tree of the optical amplifiers

象,建立了光纤放大器私有节点的树形结构。根据 MIB-II 标准建立私有节点的规则,在固定节点 Enterprise 下建立了代表光纤放大器的子节点 Opticalamplifier,它的对象标识符^[8](object identifier)为 1.3.6.4.1.24。它的子节点包括 physical feature, pump, amplifierfeature 和 signal,它们分别表示光纤放大器的物理特性、抽运参量、放大器特性和信号参量 4 个类别,用以将光纤放大器中需要被管理的对象进行分类。各个被管理对象以这 4 个类别节点为根节点在其下面都具有与自己对应的叶子节点。图 6 的树形结构中没有包含所有节点,只列举出了几个比较常用的叶子节点,cur_set 和 tenp_set 表示抽运参量类别下的半导体激光器的驱动电流和控制温度的设置值^[9],aver_gain, max_gain 和 min_gain 表示放大器特性的平均增益、最大增益和最小增益的测

试值。

管理端实现光纤放大器私有节点树形结构是根据抽象语法标记^[10](abstract syntax notation one, ASN.1)的编码规则创建了一个名为 OPTamplifier.txt 的 MIB 文本文件,管理端软件通过读取这个文本文件获得所有的节点。在代理端,光纤放大器私有节点树形结构的建立是以名为 struct mibnode 的结构体表示节点,用链表的形式把所有节点链接起来,并为这些节点分配相应大小的单片机内存空间,形成一个 MIB 表,当要存取这些节点的信息时,就可以用很简单的内存读写操作实现。

5 测试结果

将被管理的光纤放大器的模拟接口和代理端模拟数字转换(analog digit convert, ADC)接口相连,并且将管理端和代理端连接在同一个网络中,进行对象监控联合测试。代理端采集光纤放大器模拟量采用多通道多次采样取平均值的方法,并且所有对象值的数据通信都是通过以太网完成的,所以监控系统的数据传输速率和光纤通信系统线路上的传输速率无关,由于工作稳定的光纤放大器大部分对象的数据在较短的时间间隔内不会有明显的变化,所以测试结果能反映当前光纤放大器的状态,测试结果如图 7 所示。

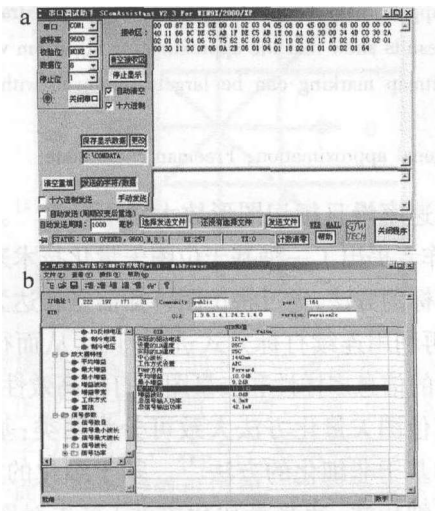


Fig. 7 The union test result

a—a complete frame b—the return information of part of objects

图 7a 中给出了代理端在正确接收到管理端单独的一种操作后,通过串行通信的方式在“串口调试助

手”软件界面接收区显示返回给管理端的一个完整的 16 进制数据帧。这个操作对应的 SNMP 包的所有信息都包含在这个数据帧内。图 7b 显示了管理端部分节点在进行相应的操作后获得的返回信息值,这些值反映了采集到光纤放大器的各项实际指标。

6 结 论

由于 SNMP 的简单性和易用性,已经成为 TCP/IP 网络事实上的管理标准,目前越来越多的具有网络性质的产品都直接提供了 SNMP 接口。作者结合实验室自行研发的光纤放大器,在 Windows 操作系统环境下实现了 SNMP 管理软件界面,在 MSP430 单片机平台上实现了 SNMP 代理功能,并编写了光纤放大器的专用 MIB 库。设计的系统已与实际光纤放大器联合测试,通过远程主机可实现对 MSP430 单片机平台控制的光纤放大器进行基本功能的监控和配置。

参 考 文 献

- [1] WANG J, CHENG Zh, LI H B, et al. SNMP agent extended to manage for telecommunication power device[J]. Telecom Power Technologies, 2008, 25(1):70-72(in Chinese).
- [2] LIU X F. Research of MIB browser on SNMP++[J]. Computer Engineering and Applications, 2009, 45(3):91-93(in Chinese).
- [3] CHENG J X. Development of SNMP based network management system by VC++ and SNMP++[J]. Modern Computer, 2007(9):127-129(in Chinese).
- [4] AYRES J. Using the crystal CS8900A in 8bit mode[R]. Austin, Texas, USA: Cirrus Logic Corporation, 2000:1-4.
- [5] STEVENS W R. TCP/IP illustrated volume 1 the protocols [M]. 2nd ed. Beijing: China Machine Press, 2005:256-269(in Chinese).
- [6] HE J F, LI Sh, LI X Sh, et al. Analysis and implementation of SNMP agent software on embedded devices[J]. Information and Electronic Engineering, 2007, 5(4):288-291(in Chinese).
- [7] McCLOGHRIE K, ROSE M. RFC1213, Management information base for network management of TCP/IP-based internets: MIB-II [S]. Mountain View, CA, USA: Internet Engineering Task Force, 1991:9-10.
- [8] ZHOU J, ZHANG X T, WANG Q. Dynamic MIB structure of SNMP and high performance search algorithm[J]. Computer Engineering, 2008, 34(2):171-174(in Chinese).
- [9] TAN Zh Q, LONG X W. Design of driving circuits of semiconductor lasers for measurement of continuous-wave cavity ring-down technology[J]. Laser Technology, 2008, 32(1):27-29(in Chinese).
- [10] ZHOU B, HUANG J W. Design and implementation of ASN.1 parser and auto BER en/decode software[J]. Communications Technology, 2007, 40(11):275-276(in Chinese).