

车间层嵌入式设备网络通信协议的研究与开发

Research and development of network communication protocols on embedded device in shop floor

童亮 鄢萍 刘飞 (重庆大学制造工程研究所 重庆 400044)

摘要:在分析了车间设备层通信中出现的网络瓶颈问题的基础上,针对网络化制造的特点,提出基于嵌入式设备的多种通信方式相结合的方法来实现其与上下层设备的通信,包括基本通信方式、RDP 方式、嵌入式浏览器方式和音视频通信方式,并针对与底层设备的通信问题,提出通信协议的优化定制方法,并开发实现得出优化结果。

关键词:通信协议 车间设备层 定制 网络制造 嵌入式设备

1 引言

在复杂的网络制造环境下,大量信息在上下层设备间流动,其中有大量的冗余数据占用了网络带宽,使车间层成为整个系统的瓶颈,造成网络的拥挤现象,而且不同的设备间有不同的通信方式,这就需要对网络系统中所涉及到的通信方式及协议进行整体考虑,把流过车间层设备的数据进行优化整合,从而用更少的中间层设备来传送更多的数据信息、控制更多的现场设备,降低制造成本。目前国内外对于网络制造环境下,在这个层面上对多种网络通信方式及协议相结合的研究还少见报导,本文在结合对制造企业所应用的网络技术深入调研的基础上,提出多种通信方式相结合的方法实现通信,并实现对车间设备层信息流的优化定制。

在这里进行处理加工的。这些设备通过现场总线及其相关协议向下层设备收发数据,同时又通过 TCP/IP 的可靠通信机制同上层交互。

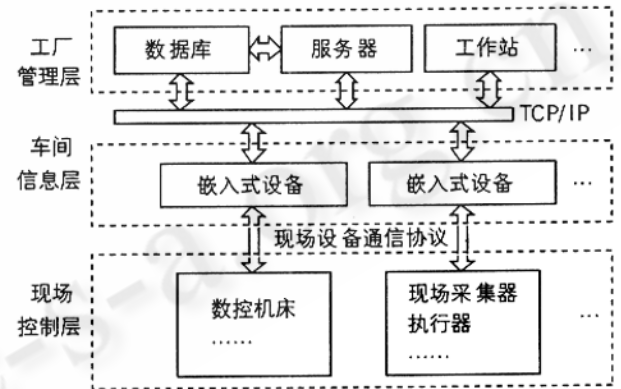


图 1 企业三层网络结构图

2 车间层的网络通信

2.1 制造环境下的网络结构

在网络制造环境下,随着网络体系结构不断更新,TCP/IP 通信技术和现场总线使 CIMS 企业各个层次^[1]的信息交互变为现实,形成了三层网络结构,如图 1。车间层与其它层及车间层内部的信息交换就成为了车间层信息管理的关键问题。

网络通信协议的层次结构决定了其硬件无关的特性,因此从数据传输的层面上看,信息都是以数据流(由数据帧组成)的形式流过车间层的嵌入式设备,并

2.2 流过车间设备层信息流的分析

下层设备的数量大,应用层通信协议主要由生产厂家定制,数据冗余大,因此需要在数据通过车间层设备时对其进行优化处理。现场采集数据可分为两类:缓变数据(如温度、湿度等)和时变数据(如电压、气压差等)。针对缓变数据实时性要求不高的特点,可采用降低采样频率的方法减小数据量。而对于时变数据,可采用“缓存上传”方法,即数据在车间层进行实时处理(显示、报警、控制等),除去数据冗余,并将数据缓存后,分时向上层发送,减小对网络的开销,如图 2。

因为在 socket 的技术规范中没有规定应用层,所

以通信双方要实现数据交互,就必须自行定制一套完整的应用层握手协议,目前较为成熟的应用层协议大都是通过对传输层数据流的定制,实现在网络上三层的握手。特别在对数据的安全性要求较高,通用协议又不能完全满足需要的场合下,自定义应用层通信协议就显得尤为重要了。上下层设备的数据交互将涉及到多种离散的通信方式及协议,对其进行统一规划,把成熟的通信协议和自己定制的协议相结合,扬长避短,使整个系统有序地运行。

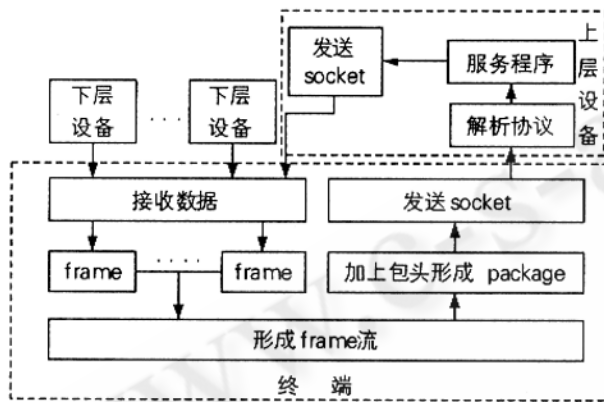


图 2 数据传送

3 通信协议的研究

3.1 基本通信方式及通信协议

网络制造环境下的基本通信方式是:车间层设备对下层设备传来的数据进行优化定制,通过 TCP/IP 协议将其发送到上层后解析;上层命令通过 TCP/IP 协议发送到车间层设备,经转化后通过现场总线下载到现场设备。

3.1.1 底层设备信息流的优化定制原则

网络协议可以简单地理解为一个由特定数据按照特定格式组织的、在特定网络系统中传输的字符串^[5]。作者总结出通信协议优化定制原则:

(1) 分层定制。根据网络分层模型,在应用层将双方的握手协议分层,实现各层间数据互不干扰。

(2) 数据结构整合。在认真分析被监控设备的通信协议的基础上,尽量对数据进行优化整合,减少数据冗余,在数据量大的时候,数据可进行压缩。

(3) 数据的加密。在数据安全性要求较高时,可

对数据按一定的算法进行加密。

(4) 数据格式统一。尽量简化设计,双方采用统一的协议格式发送和接收数据。而且双方的通信模块尽量用同种编程语言编写,以提高代码的重用率。

(5) 可扩展性。对于以后的修改和补充,需要协议是可扩展的。

遵循以上原则,把所有双方在握手时所涉及到的内容进行统一定义,最后形成比较完备的文本。

3.1.2 优化定制的具体方法

将现场各被监控设备进行编号(设备号),每台设备可能要监控多个点,对每个点进行编号(点号)。将设备一次采集的多个数据提取出来,存入优化后的结构中,定义为数据帧。帧数据表示为某台设备某时的状态,如图 3(a)。

与上层服务器端的通信是通过建立在 TCP/IP 协议基础上的套接字(socket)接口进行的。将一定时间内的多个数据帧连接成数据流,加上自定义的包头,定义为数据包。将数据包按 socket 数据流的形式发送出去。即数据包表示为所监控的设备在某段时间内的状态^{[4][5]},如图 3(b)。将包结构的数据区定义为数据流即可实现文件传送。

3.2 与数控机床的通信

现今常用的数控系统(如 FANUC 7M),包括一些老机床,其内存容量小,而借助 CAM 系统自动生成的数控程序量巨大,无法一次性加工,必须人工进行分段,不仅影响机床使用效率,还可能因人为错误造成废品^[6]。与数控机床除了读写其状态和控制信息外,还要解决复杂数控程序的断点续传问题。由于车间加工零件种类繁多,数控程序大都采用文件夹的方式保存,查找困难,不能实现权限管理。因此需要通过建立数据库对数控程序进行统一管理,克服传统文件夹管理方式的弊端,实现资源共享、程序管理和多机床同时在线加工。

对数控程序进行管理,向数控机床下传数控程序,并提供网络监控和故障诊断服务是生产制造过程中的一个重要的环节,为了提高效率,需要在车间层设备上运行守护程序与数控机床直接进行数据交互。

3.3 RDP 方式

RDP(Remote Desktop Protocol)是基于 Windows

终端服务的通信协议。集成于 Windows Server 或 Windows NT 操作系统中,支持多用户同时登录,完全使用远程网络资源,通过本地 GUI 实现 Windows 终端。RDP 能够在不同虚拟通道(Channel)中实现数据传输、设备通信和高性能数据加密,可以把本地的端口(如串口、打印机等设备)重定向(Redirection)到服务器端,

端将收到的数据包解(分层)开还原成图像。客户端事件(包括键盘、鼠标和端口读写等)被重定向到服务器上;服务器端用虚拟设备接收这些消息,并将运算后客户端需要更新区域的图像或对端口的操作发送回客户端。

通过嵌入式 RDP 客户端连接远程终端服务器,利用 Windows 强大的 GUI 功能,实现上下层设备间的直接交互,从设备层到管理层的直接通信,和企业控制管理的无缝集成。在应用程序的开发上,利用 Windows 的开发工具,大大加快软件开发的速度。

3.4 多种通信方式相结合

其它通信方式还有:嵌入式浏览器(HTTP)方式和音视频通信(H.323)方式等。

利用嵌入式浏览器实现对 Web 网页的浏览;用 3.1 节的方法定制通信协议,编写后台守护程序,实现对底层设备的监控、数据的上传和数控机床程序的下传等功能,同时在前台进行显示;通过以上两种方式的外部调用,都可以实现对 RDP 方式和 H.323 方式的集成。其中 RDP 方式用于车间任务的下达及完成情况的反馈;用 H.323 方式实现与服务器端的音视频通信。利用嵌入式设备,以嵌入式浏览器或监控程序为核心,运用 B/S、C/S 或两者相结合的模式,在车间层实现多种通信方式集成的方法,已在某机床厂工具车间进行了应用,效果良好,在嵌入式设备的高端应用上具有很高的参考价值。

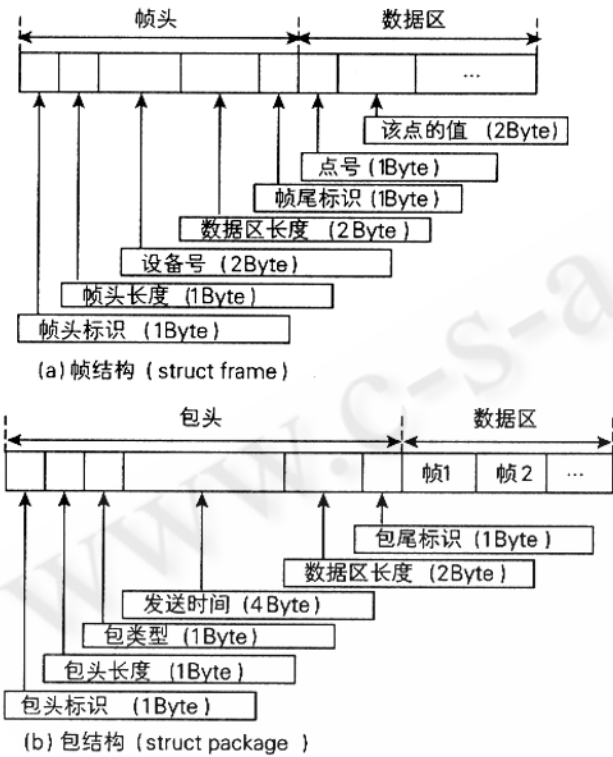


图 3 优化后帧(包)结构图

表 1 对图 3 中的解释

帧(包)头/尾标识	用户自定义
帧(包)头长度	一般是固定的。
设备号	标识本帧的内容所属的设备编号。
数据区长度	本帧(包)所包含的数据点(帧)数。
点号和点值	一个监控点的数值。
发送时间	本包发送的时间。
包类型	为协议提供扩展以及数据格式的统一,用户可通过不同的类型设定不同的数据区格式,如报警信息、呼叫信息、返回信息等。

使服务器端可以直接对底层设备进行读写操作。RDP 分层对数据进行处理,由上到下分别是 RDP 层,SEC 层,MCS 层,ISO 层和 TCP 层。在 TCP 层及其以下通过 TCP/IP 协议进行数据的传输。其工作流程是:服务器端将图形设备产生的图像压缩打包(分层)发送,客户

4 应用实现

4.1 嵌入式设备的实现

以重庆大学和重庆海特克系统集成有限公司的“车间、设备层信息交互终端”为系统平台,该平台以基于 MIPS 构架的嵌入式 SOC 和嵌入式 Linux 系列软件组件构成,能够实现 RDP、嵌入式浏览器、音视频通信等通信方式,而且只需通过 Linux 标准的 GNU 编程即可开发后台的服务程序,并用 GUI 实现人机交互界面^[7]。

4.2 网络 I/O 的实现

网络通信分为阻塞和非阻塞两种模式。对于收发数据,在阻塞模式下,进程将阻塞直到调用结束或超时出错,在单线程应用中不能使用。select 模型能同时对

多个端口进行 I/O 监控并超时返回,还能跨平台,使进程(线程)不会无限期地等待下去。所以,在嵌入式系统网络中应采用:多线程 + 阻塞模式 + select 模型。其关键部分代码如下:

```

if( select(0, &fs_Read, NULL, NULL, &m) > 0 ){
    if ( FD_ISSET( fp0, &fs_Read ) )
        pthread_create( &t, NULL, ( void * ( * ) ( void * ) )
            thd,NULL);
    if ( FD_ISSET( fp1, &fs_Read ) )
        /* ..... */
} //当 select 返回且端口 fp0 可读时,创建新线程 thd.
    
```

4.3 优化协议的应用及效果

作者根据上述方法和对底层设备通信协议的研究分析,开发出一套优化协议,具体情况如下:3 台设备通过串口同时向嵌入式设备发送数据,串口 1 和串口 2 是实变数据,串口 3 是缓变数据。用 3.1 节的方法将每个串口收到的多个数据帧进行优化整合,去掉数据冗余,表 2 为优化前后的比较。从表 2 可以看出,优化后的数据量减少了 67.36%,大大降低了对网络资源的占用。数据上传是分时传送,每次发送是将一定的时间间隔内采集的数据都发送出去。在嵌入式设备上建立一个循环使用的缓冲区,数据按帧依次存入缓冲区。定义两个指针:一个指向下一次发送的起点(即上一次发送的终点);另一个指向动态指向缓冲区的终点,并且随数据的采集而向后移动。每次上传两个指针之间的数据。向服务器传送数据时,每隔 300ms 传送一次,每个数据包的长度约为 440(包头 60)字节。

表 2 协议优化前后比较

	比特率	优化采集频率	每秒上传数据量		优化率
			优化前	优化后	
串口 1	19200	10Hz	60 帧 × 5 B	约 10 帧 × 12 B	60.00%
串口 2	115200	20Hz	200 帧 × 17 B	约 10 帧 × 115 B	61.18%
串口 3	9600	0.5Hz	4 帧 × 50 B	约 0.5 帧 × 6 B	98.50%
合计			3900B	1273B	67.36%

4.4 实现对数控程序的管理

(1) 基本通信方式。数控程序是以文件形式存放在服务器上的,编写基于嵌入式 Linux 系统的多线程守护程序:一方面,用 socket 接口从服务器端得到加工程

序列表,工人根据加工需要选择数控程序,系统为每个加工机床创建一个线程,并对其信息进行管理,利用 3.1 节的方法定制通信协议,实现上传请求和下传分段程序;另一方面,采用 select I/O 方式实现对多个机床通信端口的侦听,得到每台机床的状态信息,根据加工进度,并按照机床的通信协议,自动将其分段程序下载到机床的存储器中(如图 4)。其中,“select 侦听 + 多线程管理”和“数控程序分段管理”是整个系统的核心。

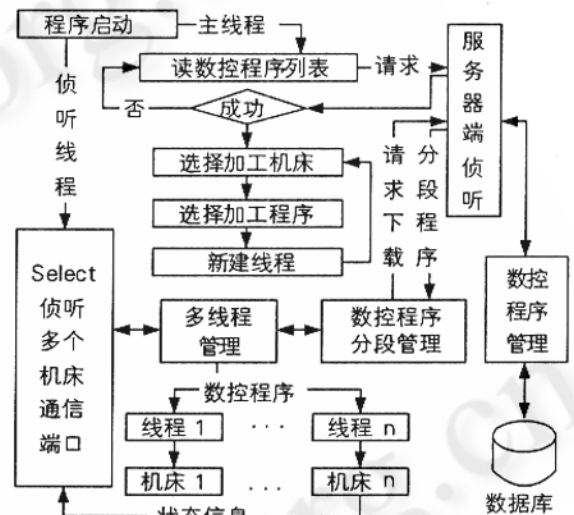


图 4 数控程序管理系统

在主线程中加一个定时器,按一定的时间间隔向每台机床发送状态信息请求,并在每个监控线程中接收并显示。

在每个线程中保存数控程序分段信息(包括文件名、断点位置和分段长度等),当机床请求或需要断点续传时,根据分段信息从服务器上取得下一段加工程序的代码,传给机床。

本系统实现了数控机床网络化,彻底摆脱了手工或单机录入程序的局面。当生产任务紧张,需要多台机床加工同一任务时,该系统可以提供多机床同时在线加工的功能,而且允许程序的断点续传加工,提高了生产效率。

(2) RDP 方式。在嵌入式设备上用 RDP 客户端登录到服务器,实现 Windows 虚拟机,并将嵌入式设备的通信端口重定向到虚拟机上。如图 5,除了端口、显

示设备和鼠标键盘之外,在中间处理数据的过程对用户完全透明,用户就像是独自使用一台连接有底层设备的 Windows 终端。

Windows 提供了一组 API 函数对串口进行操作,它们是:

```
CreateFile("COM1", .....); //打开串口
CloseHandle(); //关闭串口
SetupComm(); //设置收发缓冲区
SetCommState(); //设置串口参数
ReadFile(); WriteFile(); //读、写串口
```

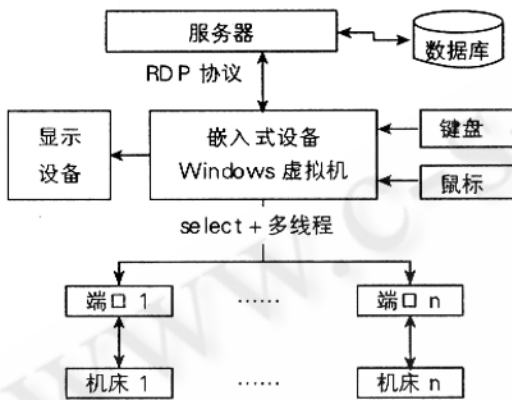


图 5 基于 RDP 的数控程序管理系统

此外,也可以利用第三方控件进行开发,如基于 ActiveX 技术的 MSComm 控件,基于 VCL 的 SPComm 构件等,由于其封装好,受到广大编程人员的青睐。

通过开发基于 Windows 的应用程序读写本地端口,实现直接与机床通信。采用 select I/O + 多线程的方式实现同时对多个机床通信端口(机床)的监控。利用“数控程序分段管理”实现断点续传,并通过维护数据库来管理数控程序。与上节相比,RDP 方式隐藏了与服务器的通信过程,而且 Windows 上的开发工具功能强大,减少了开发人员的工作量。

5 结论

论文分析了车间层设备的通信方式及协议,根据嵌入式网络开发经验,在兼顾到易于开发、灵活易用的基础上,提出了多种通信方式相结合来实现设备间信息交互,并通过对信息流的优化定制,解决了车间设备层的网络瓶颈问题,在网络制造环境下有着普遍实用

意义,在工程实践中具有很高的参考和借鉴的价值

参考文献

- 1 刘国栋、孙增圻、刘化君等,异构环境下车间层信息集成的分析与研究,信息与控制,2001,30(6):522-525 [英]LIU Guodong,SUN Zengqi,LIU Huajun, et al. Analysis and study of information integration in shop floor in iso meric environment. Information and Control,2001,30(6):522-525.
- 2 NAKAJIMA T. Implication of embedded Linux in Japanese embedded industries. Object - Oriented Real - Time Distributed Computing,2003. Sixth IEEE International Symposium on 14 - 16 May 2003 Page(s): 41 - 43.
- 3 NAVET N, SONG Y, SIMONOT - LION F, WILWERT C. Trends in automotive communication systems. Proceedings of the IEEE, Vo. 93, No. 6, June 2005 Page (s):1204 - 1223.
- 4 K. F. Tsang, L. T. Lee. A novel communication protocol for wireless short command. Consumer Electronics. IEEE Transactions on Consumer Electronics. Vol. 49, No. 4, November 2003 Page(s):1020 - 1027.
- 5 王承君,自定义串口数据通信协议的分析与设计,计算机工程,第 30 卷 24 期,2004 年 12 月:192 - 194 [英]WANG Chengjun. Analysis and Design of Self - defining Serial Port Communication Protocol. Computer Engineering, Vol. 30 No24, December 2004: 192 - 194.
- 6 田原,数控加工程序分段软件设计,煤矿机械,Vol. 27 No. 4,2005 年 4 月:562 - 563 [英]TIAN Yuan. SoftWare for Disintegrating a CNC Program. Coal Mine Machiner, Vol. 27 No. 4, April 2005: 562 - 563.
- 7 鄢萍、刘飞、贺德强等,网络化制造中的一种多功能信息终端. 机械与电子,2004,5:3 - 5 [英]YAN Ping,LIU Fei,HE De qiang, etc. A Kind of Multi function Information Terminal in Networked Manufacturing. Machinery & Electronics, May 2004?:3 - 5.