轻量级嵌入式TCP/IP协议栈的设计

介绍了引入轻量级TCP/IP协议栈的背景,描述了标准TCP/IP协议栈,分析了嵌入式系统中TCP/IP协议栈的特点和设计的思路,总结了轻量级TCP/IP协议栈和标准TCP/IP协议栈的区别。介绍了引入轻量级TCP/IP协议栈的背景,描述了标准TCP/IP协议栈,分析了嵌入式系统中TCP/IP协议栈的特点和设计的思路,总结了轻量级TCP/IP协议栈和标准TCP/IP协议栈的区别。嵌入式系统　轻量级　TCP/IP协议栈计算机工程王力生　梅岩　曹南洋同济大学计算机科学与工程系,上海2000922007第六图书馆

第33卷 第2期 计 算 机 工 程 2007年1月
Vol.33 No.2 Computer Engineering January 2007
·开发研究与设计技术· 文章编号：1000—3428(2007)02—0246—03 文献标识码：A 中图分类号：TP311

轻量级嵌入式TCP/IP协议栈的设计

王力生，梅 岩，曹南洋
(同济大学计算机科学与工程系，上海 200092)

摘 要：介绍了引入轻量级TCP/IP协议栈的背景，描述了标准TCP/IP协议栈，分析了嵌入式系统中TCP/IP协议栈的特点和设计的思路，总结了轻量级TCP/IP协议栈和标准TCP/IP协议栈的区别。

关键词：嵌入式系统；轻量级；TCP/IP协议栈

Design of Light-weight TCP/IP Stack in Embedded System

WANG Lisheng, MEI Yan, CAO Nanyang
(Department of Computer Science and Engineering, Tongji University, Shanghai 200092)

【Abstract】 The paper introduces the light-weight stack. It describes the standard TCP/IP stack. It gives emphasis to the characters of TCP/IP stack in embedded system and design methods. A comparison between standard TCP/IP stack and light-weight TCP/IP stack is listed.

【Key words】 Embedded system; Light-weight; TCP/IP stack

1 概述

嵌入式Internet是近年来随着嵌入式系统的广泛应用和计算机网络技术的进步而发展起来的一项新兴概念和技术。目前，以单片机或微控制器(MCU)构成的嵌入式系统已被广泛应用于家庭、工业、军事等各个领域，在应用数量上远远超过了通用计算机。而随着当前Internet技术的迅猛发展，各种嵌入式系统也产生了网络互联的需求。整个地球将会披上一层无所不在的“电子皮肤”。要想实现嵌入式系统的Internet网络化，就必须在嵌入式系统中实现TCP/IP协议栈。这已经成为嵌入式应用领域和网络互联领域的研究热点。

要实现嵌入式TCP/IP协议栈，最关键的一个问题就是，如何结合系统资源有限的嵌入式系统软硬件环境以及具体的嵌入式Internet应用对标准的TCP/IP协议栈进行简化，从而实现一种“轻量级”的TCP/IP协议栈。

在轻量级TCP/IP协议栈方面，国内外做了许多研究。Jeremy Bentham的PICmicro协议栈只需要几十个字节就可以运行一个能够提供简单网页服务的Web server。PICmicro协议的实现面向Web server应用，与轻量级Web server应用程序紧密耦合。它的实现建立在许多应用以及系统环境的假设上。比如，它在TCP/IP协议栈中没有实现重传机制，而是假设与之通信的主机支持实现了重传机制的标准TCP/IP协议栈，在发生错误或者数据包丢失的情况下，与之通信的主机会不断地发送重传请求来保证TCP协议的可靠性。

瑞士计算机科学院的Adam Dunkels写的uIP可以说是一个针对嵌入式系统特别是8位或16位嵌入式系统的轻量级TCP/IP协议栈，它以库函数的形式提供给嵌入式Internet应用开发人员，并采用了一种基于事件驱动的程序模型来减少代码容量和RAM的占用量。但是uIP协议栈实现的结构比较混乱，也没有包含UDP协议。

同样也是瑞士计算机科学院开发了另外一套用于嵌入式系统的TCP/IP协议栈——LwIP(Light Weight，轻型)。LwIP可以移植到操作系统上，也可以在无操作系统的情况下独立运行。LwIP实现的重点是在保持TCP/IP协议主要功能的基础上减少对RAM的占用，这使得LwIP协议栈很适合在低端嵌入式系统中使用。LwIP有如下特性：支持多网络接口下的IP转发；支持ICMP协议；包括实验性扩展的UDP；包括阻塞控制，RTT估算和快速恢复/快速转发的TCP；可选择的类似Berkeley的socket API；支持DHCP；支持PPP；以太网的ARP。

2 通用TCP/IP协议栈简介

TCP/IP协议层次结构如图1所示。

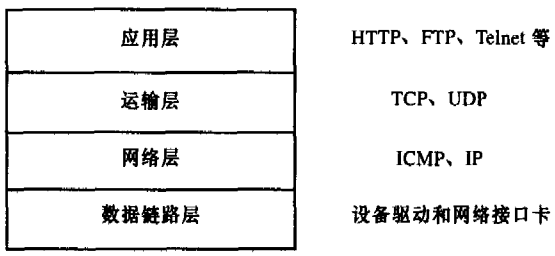

图1 TCP/IP协议簇的4个层次

TCP是为同一个网络上的计算机之间进行点到点通信而设计的，其详细说明在RFC793中可以找到；而IP是为连接在不同网络或者WAN上的计算机之间能够相互通信而设计的，其详细说明在FC791中可以找到。自从TCP/IP在20世纪70年代早期被引入之后，该协议已经被广泛使用在全世界的网络上。在PC、UNIX工作站、小型机、Macintosh计算机、大型机以及用于连接客户机和主机的网络设备上都可以使用TCP/IP。通过TCP/IP，成千上万个公共网络和商业网络连接到了Internet上，使得大量用户可以对之进行访问。TCP/IP

作者简介：王力生(1956－)，男，副教授，主研方向：嵌入式系统及其应用；梅 岩、曹南洋，硕士

收稿日期：2006-03-26 **E-mail**：powermei1981@hotmail.com

也是一种分层协议，这一点与OSI协议层次有些类似，但是并不完全相同。TCP/IP大约包含近100个非专有的协议，通过这些协议，可以高效和可靠地实现计算机系统之间的互连。

3 嵌入式TCP/IP协议栈的特点

随着嵌入式系统与网络的日益结合，在嵌入式实时操作系统中引入TCP/IP协议栈，以支持嵌入式设备接入网络，成为嵌入式领域重要的研究方向。但是传统的TCP/IP协议实现在实时性方面做得不够好，而是把大量的精力花在保证数据传送的可靠性以及数据流量的控制上。在实时性要求比较高的嵌入式领域，传统的TCP/IP并不能满足其实时要求；另外传统TCP/IP的实现过于复杂，需占用大量系统资源，而嵌入式应用的系统资源往往都很有限。因此，需要把传统TCP/IP在不违背协议标准的前提下加以改进实现，使其实时性得到提高，占用的存储空间尽可能少，从而满足嵌入式应用的要求。嵌入式TCP/IP协议栈通常需要具备以下几个特点：

(1)可裁剪。由于嵌入式应用的要求千差万别，各种嵌入式应用对网络系统的要求也不尽相同，且嵌入式系统对产品的成本、功耗比较敏感，因此嵌入式TCP/IP网络系统必须提供可裁剪的机制，能根据嵌入式系统的功能要求选择所需的协议，对完整的TCP/IP协议簇进行裁剪，以满足用户的需要。

(2)采用“零拷贝”技术提高实时性。由于TCP/IP协议的层次特性，每个协议层次都有自己的数据格式。发送数据时，各个协议层从自己的上层协议接收数据，加上本层的控制信息后再交给自己的下层协议，即打包。接收数据时，各个协议层从下层协议接收数据，取出本层的控制信息后再把剩下的数据交给上层协议，即拆包。用户数据在从本地主机传输到远地主机的过程中，需要不断地打包和拆包。如果各个协议层之间均采用数据拷贝进行数据传输，则将大大增加系统开销，从而降低系统性能。在嵌入式TCP/IP网络系统中，普遍采用“零拷贝”技术以解决该问题。所谓“零拷贝”，是指TCP/IP协议栈没有用于各层数据传输的缓冲区，协议栈各层之间传递的都是数据指针，只有当数据最终要被驱动程序发送出去或者是被应用程序取走时，才进行真正的数据搬移。

(3)可扩展。由于嵌入式系统的多样性，使得网络接口多样化、网络应用多样化，这就要求TCP/IP网络系统提供便于扩展的网络接口，方便不同驱动程序和网络应用的开发；并且根据实际的需要，可以添加新的协议栈模块到TCP/IP网络系统中，实现对新的网络协议的支持。

(4)采用静态分配技术。如果在网络发送或接收的过程中，某一次传送的数据超过了在一个物理网络上能够传输的最大数据量(MTU)，则处理该数据的任务往往会阻塞等待，直到上层重新调整需要处理的数据量的大小，它才会继续执行下去。嵌入式TCP/IP网络系统通常采用静态分配技术，在网络初始化时就静态分配通信缓冲区，设置了专门的发送和接收缓冲(其大小一般小于或等于物理网络上的MTU值)，从而确保了每次发送或接收时处理的数据不会超过MTU值，也就避免了数据处理任务的阻塞等待。

4 轻量级嵌入式TCP/IP协议栈的设计思路

对标准TCP/IP协议栈进行简化必须结合其运行的系统环境以及特定的嵌入式TCP/IP应用。这也是由嵌入式系统的内在特性决定的，嵌入式系统的特点就是面向特定的应用，软硬件可裁剪，以满足系统在体积、功耗、可靠性、成本等方面的特定要求。

4.1 底层系统运行环境

由于嵌入式系统在计算资源和存储资源方面的局限性，运行于其上的TCP/IP通信协议栈直接与硬件进行交互，没有实时多任务操作系统的支持。程序结构一般采用前后台系统——顺序执行和硬件中断相结合的方式。在前后台系统中，TCP/IP协议栈一般作为应用程序的一部分加以实现，没有标准的实现形式可遵循，应用程序设计人员可根据特定的系统环境和应用的需要自由地选择。而在有操作系统支持的应用系统中，TCP/IP协议栈一般作为操作系统内核的一部分实现。利用操作系统提供的进程(线程或任务)抽象可以把TCP/IP协议栈划分成几个进程(线程或任务)。应用程序和TCP/IP协议栈的交互通过标准的BSD socket应用程序接口。

在前后台系统中，处理器的时钟频率低，地址、数据总线窄，对一个数据包的处理要花一定的时间，势必要影响其他中断和任务的执行。因此，对TCP/IP协议栈部分的处理一般放在主程序顺序循环中，对物理网络接口控制芯片一般采用查询方式(也可采用中断的方式)，在其他中断任务的执行间隙进行TCP/IP协议栈的处理，以牺牲响应时间来换取系统可靠性和效率。

4.2 缓冲区管理

在没有计算资源和存储资源限制的计算机系统中，TCP/IP协议栈中缓冲区大小的选择可以不考虑内存的大小问题。如Unix系统中TCP/IP协议栈使用mbuf数据结构来构成一个存储链表，这个链表可以根据需要动态地增加和减小。同时运行这些TCP/IP协议栈的系统拥有相对较快的处理速度，底层物理网络接口电路中的接收缓冲区几乎不可能发生溢出和丢包的情况。而在8位或16位嵌入式系统中存储空间的数量级一般只有几十个千字节，因此在这些系统中设计和实现TCP/IP协议栈，不但要考虑TCP/IP协议栈本身的大小，还要考虑缓冲区的设置。考虑到一个最大以太网数据帧有1 500多个字节，而嵌入式系统中的存储空间十分有限，并要被各个协议所共用，可以选择申请一个能存放最大以太网数据帧的缓冲区来存放接收到的数据帧；收到一帧处理一帧，其存储地址是固定的，不是根据需要动态分配的。同时这个缓冲区还可以作为发送数据帧时的缓冲区。

4.3 IP层

TCP/IP协议栈中的网络层主要进行路由、分片、重装、向相邻协议层进行数据包的传递等处理。一般情况下，8位或16位的嵌入式系统都是作为瘦客户端或瘦服务器端。因此在这些系统中实现TCP/IP协议栈时可以把协议中的路由功能裁剪掉。对于网络层中分片和重装等功能，根据嵌入式系统的特点和特定的环境，可以对系统底层的通信环境做出一些假设，使其能在大多数情况下正常工作即可。例如，不同的底层物理网络所支持的数据帧的格式以及最大传输单元(MTU)是不同的，如果一个IP数据包在路由的过程中经过最大传输单元不同的物理网络，那么IP数据包目的端的网络层必须支持IP数据包的分片与重装等功能，否则，TCP/IP协议栈将不能正常工作，而实际现有的底层物理网络一般都是以太网，在这种情况下可以不考虑IP数据包的分片和重装问题，这种经过了简化的TCP/IP协议栈是可以正常工作的。

4.4 TCP层

TCP/IP协议栈中主机到主机传输层提供面向连接的、端到端的可靠通信服务。在这一层中，滑动窗口协议、流量控

制、拥塞控制、TCP 连接状态、往返时间估计、重传等都是比较重要的机制。为了提高 TCP/IP 协议栈的性能，没有计算资源和存储资源限制的系统中 TCP/IP 协议栈一般都实现了滑动窗口协议。对 8 位和 16 位嵌入式系统来说，实现滑动窗口协议显然不合适。前面提到在 8 位和 16 位嵌入式系统中，TCP/IP 协议栈缓冲区大小设置为可以存放一个最大以太网数据帧。在进行数据传输时采用“发送－停止－等待－确认－发送”的方式。由于不实现滑动窗口协议，因此 TCP/IP 通信协议每次只缓冲和发送一个包，接收一包处理一包，在 8 位和 16 位嵌入式系统中流量控制和拥塞控制也不需要考虑。嵌入式 Internet 中的 8 位和 16 位嵌入式系统一般作为服务器端，在这些系统中实现主机到主机传输层中的 TCP 有限状态机时，可以对一般的 TCP 有限状态机进行简化。主机到主机传输层中的往返时间估计、定时重传等机制用来保证数据包在路由过程中发生丢失时能保证 TCP/IP 协议栈还能正常工作，同时又保证 TCP/IP 协议栈的性能。基于嵌入式 Internet 中的 8 位和 16 位嵌入式系统一般用于服务器端，与之通信的一般都是实现了完整的 TCP/IP 通信协议的设备。当 8 位和 16 位嵌入式系统发送的包丢失后，与之通信的一端在确定的时间内没有收到确认包，就会认为自己所发的包丢失，超时重发。因此，8 位和 16 位嵌入式系统中 TCP/IP 协议栈的主机到主机传输层可以不实现往返时间估计、定时重传等机制。

5 小结

综上所述，可以看出应用于 8 位和 16 位嵌入式系统中的轻量级 TCP/IP 协议栈和一般操作系统的 TCP/IP 协议栈的主要区别如表 1 所示。

表 1 轻量级 TCP/IP 协议栈和标准 TCP/IP 协议栈的比较

	嵌入式系统中的轻量级 TCP/IP 协议栈	标准 TCP/IP 协议栈
底层系统运行环境	直接面对硬件，没有实时多任务 OS 的支持，程序结构一般采用前后台——顺序执行和硬件中断相结合的方式	有多任务 OS 支持，以分时为基础，多进程(线程或任务)并发执行
缓冲区管理	申请一个固定的缓冲区(一个最大以太网数据帧的大小)，收到一包处理一包	不需要考虑内存大小问题，采取动态分配和静态分配相结合的方式
IP 层	可以不考虑 IP 数据包的路由功能，也可以不实现 IP 数据包的分片和重装功能	包含路由、分片、重装等功能
TCP 层	每次只缓冲和发送一个包，可以不实现滑动窗口协议、流量控制和拥塞控制，可以不实现往返时间估计、超时重发	实现滑动窗口协议、流量控制、拥塞控制、往返时间估计、超时重发等，实现完整的 TCP 有限状态机
接口	TCP/IP 通信协议提供的通信服务函数作为库函数形式实现，应用协议(程序)通过调用这些函数实现网络通信	作为操作系统的一部分实现，一般遵循 BSD socket 接口标准

参考文献

1 Behrouz A F, Sophia C F. TCP/IP 协议簇 [M]. 第 2 版. 谢希仁, 译. 北京: 清华大学出版社, 2004.

2 卿立军. 嵌入式 Internet 中轻量级 TCP/IP 协议栈的研究与实现[D]. 长沙: 湖南大学, 2004.

3 吴艳光. 嵌入式 TCP/IP 协议栈设计方法的研究[D] 太原: 太原理工大学, 2004.

4 Adam D. Design and Implementation of the LwIP TCP/IP Stack[Z]. Swedish Institute of Computer Science, 2001.

5 Jeremy B. 嵌入式系统 Web 服务器: TCP/IP Lean[M]. 陈向群, 译. 北京: 机械工业出版社, 2003.

(上接第 245 页)

A/D 模块通过 A/D 转换对温度等模拟量进行采样。本设计中，要对陀螺、加速度计及 IF 板的温度进行监测。测温电路采用 AD708 芯片，原理如图 6 所示。本设计中测温范围为 -40°C ~ +85°C，满足系统监测和高低温实验的要求。

3 导航计算机的软件及工作时序

TMS320VC33 在编程时可以通过 C 语言、汇编语言编程，也可以通过 C 语言与汇编语言的混合编程。具有在线编程功能，使制导系统在软件修改和维护时方便、快捷。

导航计算机的软件系统的任务包括数据采集与预处理、姿态解算、速度与位置解算、初始对准和数据发送等。

根据捷联计算的性质与导航要求的不同，以及减小计算机负担的需要，通常将计算划分成不同的周期。考虑到本系统的计算机资源有限，因此将整个计算只划分为 2 个周期。

T_1 为采样周期，采样周期需要足够高来减小频谱的混叠，在本系统中 T_1 为 1/2000s。陀螺采样后的数据经过滤波后存储起来，用于圆锥补偿。加速度计的数据则直接求和。

T_2 为解算周期，取为 0.01s。每隔 T_2 的时间间隔完成以下的计算工作：(1)利用存储的 20 个陀螺数据求和并进行圆锥补偿；(2)姿态四元数更新；(3)速度、位置更新；(4)姿态速率的提取；(5)将速率、位置等信息发送至上位机。

在本设计中，里程计用来对速度进行补偿，可有效提高惯导精度。系统完成捷联惯导计算，并将导航信息通过串口输出。

4 结束语

本文以 DSP 芯片 TMS320VC33 为核心，利用 CPLD 技术设计的捷联导航计算机系统已研制完成，从已完成的工作和运行结果来看，数据处理模块完成一次捷联基准的运算耗时小于 1ms，TMS320VC33 的 32 位字长也保证了系统计算的精度，导航计算机与外部设备的通信正常。本文设计的整个系统达到了预期目标，具有相当的实时性和较高的精度，而且具有体积小、功耗低等优点。

参考文献

1 陈　哲. 捷联惯导系统原理[M]. 北京: 宇航出版社, 1986.

2 王　勇, 刘光灿. DSP 在捷联惯性制导技术中的应用[J]. 微处理机, 2003, 24(1): 62-64.

3 张雄伟, 陈　亮, 徐光辉. DSP芯片的原理与开发应用[M]. 第 3 版. 北京: 电子工业出版社, 2003-02.

4 朱铭锴, 赵　勇, 甘　泉. DS 应用系统设计[M]. 北京: 电子工业出版社, 2002-10.

5 赵曙光, 郭万有, 杨颂华. 可编程逻辑器件原理、开发与应用[M]. 西安: 西安电子科技大学出版社, 2000-08.

6 单茂华, 周百令. 基于 CPLD 技术的大动态范围高速数据采集系统设计与实现[J]. 仪表技术与传感器, 2003, (3): 27-29.