

嵌入式 IPv6 协议栈的设计与实现^①

刘桢光, 张 品

(杭州电子科技大学 通信工程学院, 杭州 310018)

摘 要: 提出了一种基于平台无关性的嵌入式 IPv6 协议栈, 设计了相应的缓冲区管理机制, 简化了 IPv6 模块中 ICMPv6, 邻居发现协议以及 TCP 模块, 并对协议栈接口层和传输层作了相应的功能扩展。在 ARM 开发平台上对协议栈进行了测试, 运行良好。

关键词: 嵌入式; IPv6; ICMPv6; 邻居发现

Design and Implementation of Embedded IPv6 Protocol

LIU Zheng-Guang, ZHANG Pin

(Telecommunication Engineering College, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: An implementation of embedded IPv6 stack independent of any platform is proposed. Based on the embedded application, the management of buffer is designed by Mbuf technology. and some functions of ICMPv6 and Neighbor Discovery modules in IPv6 are reduced. TCP and UDP modules are extended. The stack is tested on the s3c2440 platform. All these show a good performance.

Keywords: embeded; IPv6; ICMPv6; neighbor discovery

随着嵌入式设备的应用日益广泛, 尤其在大量的嵌入式设备接入 Internet 后, IPv4 网络的问题日益突出。如 IP 地址匮乏、路由表庞大, 对安全性和移动性支持不足, 地址配置复杂, 缺乏对服务质量的支持等, 所有这些都制约着互联网的发展。IPv6 技术的出现恰好解决了 IPv4 地址空间有限无法满足数量庞大的嵌入式设备接入网络的问题, 巨大的地址空间使得每个嵌入式设备都可以作为一个独立的实体接入互联网, 不再受地址匮乏的限制。在互联网即将从 IPv4 过渡到 IPv6 的大背景下, 在嵌入式设备上实现 IPv6 是大势所趋。

1 协议栈的设计要点

1.1 协议栈的工作模型

嵌入式系统由四个层次组成它们分别是硬件资源层、操作系统层、网络协议层和应用实例层。硬件资源层必须包含基本的硬件器件, 如微处理器、RAM、

ROM、网络芯片等。嵌入式操作系统完成对硬件资源的管理、为高层应用提供良好的软件平台。IPv6 网络协议就是要使嵌入式系统能直接接收来自于 IPv6 网络的信息, 和向 IPv6 网络中发送信息, 完成与 IPv6 协议的网络通信功能。应用程序是嵌入式系统的最终目标所在, 它可以是对设备的控制, 也可以是对实时参数的获取处理等^[1]。

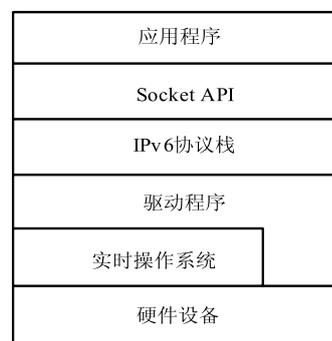


图 1 嵌入式系统的分层结构

① 收稿时间:2010-09-17;收到修改稿时间:2010-11-06

1.2 协议栈的分层结构

系统的分层按照简化的 OSI 四层模型，具体的分层情况如图 2 所示。

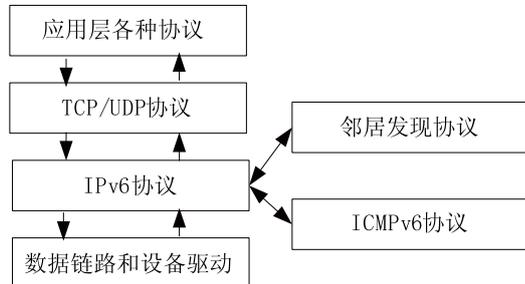


图 2 协议栈分层

数据链路层主要是提供一个网卡驱动的模板，用统一的接口函数实现上层协议需要的过程^[2]。网络层一方面完成对 IPv6 数据包基本报头的封装发送或者是接收拆包过程，并根据不同的包头内容进行不同的处理。传输层实现 TCP 和 UDP 协议的功能。应用层可以是面向具体应用的软件，嵌入式系统中的应用程序可以直接访问到协议栈内部的缓冲器和存储器的值，提高嵌入式系统的资源利用率。

1.3 平台无关性

考虑到嵌入式系统的资源有限性和以后协议移植的方便性和单进程模型，我们决定采用进程模型如下：在操作系统基础上实现一个操作系统模拟层，实现基本的时钟，消息管理，进程同步等基本操作系统功能，协议栈进程对操作系统功能的使用是通过模拟层提供的接口来实现，协议栈进程运行在用户空间，这样实现了与操作系统分离，解决操作系统依赖性。协议进程方面，把所有的协议栈协议封装到单独进程中，应用程序可以驻留在其中，也可以作为一个单独的进程来实现，协议栈作为一个单独进程解决了层间进程切换带来的时间代价，同时它也带来另外一个问题，运行在用户空间的协议栈作为一个单独的进程，它必须和其他进程一样等待着操作系统的调度^[3]。

1.4 协议栈缓冲管理

协议栈有一块专属的存储缓冲区，协议栈对它采用了简化了的 mbuf 结构单元串连起来，协议栈对协议层缓冲区进行静态和动态管理。

静态管理：事先分配部分固定大小的 PBUF POOL 类型存储单元，固定大小而且事先已经被分配好，用

于快速存储从网卡接收缓冲区传过来的数据，避免因网卡缓存溢出而丢失网络数据，降低包丢失率。

动态管理：可以动态的分配和回收 PBUF RAM 类型存储单元，不固定大小，可动态修改和分配协议栈的内部缓冲区，我们对数据使用零拷贝技术，所谓“零拷贝”技术指数据在协议栈缓冲区内部进行层间移动时不进行拷贝移动，只传递数据指针，只有当数据真正从协议栈缓冲区移走到应用程序缓冲区或者数据从网卡缓冲区进入协议栈缓冲区才执行数据拷贝。

1.5 协议栈的通讯机制

本文中的系统使用消息(Message)、信号量(Semaphore)和共享内存(SharedMemory)三种方式来完成协议栈内部及协议栈同应用程序之间的通讯任务^[4]。

2 协议栈的实现

2.1 网络接口层

网络接口是连接协议层和物理硬件的中间层，向上屏蔽具体硬件实现细节，向下通过驱动程序和硬件打交道。对各种不同的网络芯片进行控制，把从协议栈的网络层接收来的数据传递给网卡发送出去。流程如图 3 所示：

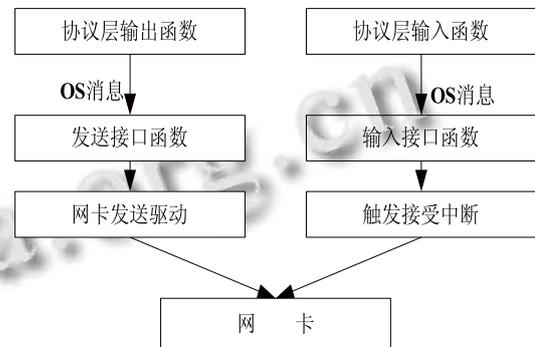


图 3 网络接口模型

接口函数介绍如下：

(1) 初始化函数

初始化函数 Ethemet Init0 负责实现网络接口层以及下层网卡的初始化。主要包括 MAC 地址的配置、以太网控制器的上电复位、收发缓冲环首尾地址的配置以及 DMA 的初始化和收发数据格式的定义等。

(2) 发送函数

网络接口层通过 Ethemet Output()l 函数完成对网络层协议产生的数据的封装，将其封装成以太网数据

帧, 并将其传送到网卡的发送缓冲中, 最后将该数据帧发送到网络。

(3) 接收函数

网络接口层通过 `Ethernet_Input0` 函数接收下层以太网卡收到的数据帧。它会根据以太网帧的类型选择不同的函数进行处理, 并去掉以太网帧头, 将正确的 IPv6 数据包从网卡的数据缓冲区发送至系统的接收缓冲区内^[5]。

2.2 IPv6

IPv6 模块主要包括两部分的功能: 一是负责从网络接口层收取 IPv6 数据包, 对数据包进行一定的处理后将数据包发送给不同的处理模块(TCP 或 ICMPv6)。二是从上层接收数据, 进行报文的选路, 封装 IPv6 报头后将数据包发送给网络接口层。涉及的主要函数有接收数据包函数 `ip_input()`、发送数据包函数 `ip_output()`。

2.3 ICMPv6 模块

ICMPv6 协议属于网络层协议, ICMPv6 报文用于 IPv6 节点对当前网络运行状况的诊断和网络信息的通告, 主要功能是接收、处理、发送 ICMPv6 报文。

首先在 `ip_input()`, 程序通过 IP 报头的下一跳报文判断报文是否为 ICMPv6 报文, 如果是则把报文交给 `icmpv6_input()` 函数进行处理^[6], 该函数首先判断报文类型, 主要有四种类型的报文: 错误报文, 未知类型的报文, 邻居发现报文, 信息报文, 对不同报文类型 ICMPv6 对它进行不同的处理, 如果为错误报文, 则函数根据 ICMPv6 报文的类型和错误代码, 把错误报文映射到一个差错代码集上, 并把报文和差错代码集

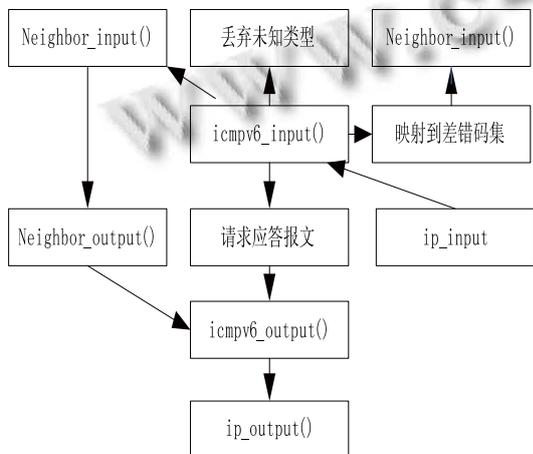


图 4 ICMPv6 流程图

上的代码交给上层协议, 上层协议再根据错误的原因进行相应的处理, 如果为未知类型, 则直接丢弃, 如果为归属于邻居发现模块的邻居公告, 邻居请求报文等邻居信息报文, 则交给邻居输入处理模块函数, 如果是信息报文, 则由 ICMPv6 模块自己进行相应处理。如图 4 所示。

2.4 ND

邻居发现 (Neighbor Discovery) 协议是 IPv6 基本协议栈中的核心协议, 利用邻居发现功能, 系统可以找到该链路上的其它主机和路由器, 本协议栈只实现了邻居请求报文和邻居公告报文, 以及基于这两种报文的地址解释, 可达性检测功能^[7]。

2.5 TCP

TCP, UDP 对 IPv6 和 IPv4 报文的处理过程没有任何的不同。但是由于地址结构的不同, 需要对 TCP, UDP 模块中使用 IP 地址的部分进行扩展。

数据结构扩展: PCB 是用来记录每个 TCP 或 UDP 连接状态的, 包括本地 IP 地址和远程 IP 地址, 扩展时使用 union 结构添加 IPv6 地址, 同时添加了标识位区别 IPv4 和 IPv6 连接^[8]。

操作扩展: 因为对 IPv4 和 IPv6 报文调用相同的 TCP 和 UDP 处理函数, 因此系统在实现时对在这几个函数的处理过程中加入对相应 PCB 中标识位的判断, 如果为 IPv6 连接, 则调用相应的 IPv6 的一系列函数。

3 协议栈的测试

本文采用的嵌入式系统基于 Samsung 公司的 s3c2440 微处理器, 开发板上移植了 Linux 操作系统, 在 PC 机上可以通过 ping6 命令来检测嵌入式 IPv6 协议栈的运行情况, 结果证明此协议栈运行稳定、正常。

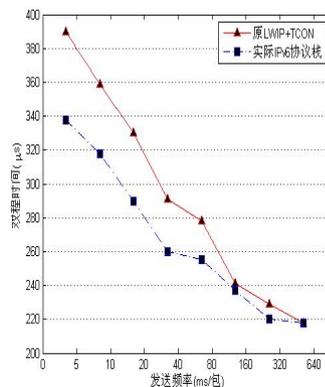


图 5 实时性能对比

如图5所示为LWIP和IPv6协议栈实时性实验对比图,本论文设计的IPv6协议栈在网络负载较轻时(每间隔640ms发送1个TCON包),具有很好的实时相应性能(213us),网络负载较重时(每间隔5ms发送1个TCON包),也能达到较高的实时响应性能(334us)。并且基于IPv6的TCON包比基于IPv4的TCON包实时性能要好,这在网络负载重时(小发送频率)尤为明显。实验证明本协议能够满足较好的实时性需求。

4 结语

本文主要阐述了一种嵌入式协议栈的设计与实现,讨论了协议栈的一些精简技术,在此基础上对IPv6的协议栈进行了裁剪,实现了基本的IPv6通信功能,扩展IPv6的移动功能和IPv6地址自动配置是此协议栈下一步的研究重点。

(上接第183页)

```
v_formula:=          f_replace_para_with_value
(v_formula,v_para_id,v_para_value);
end loop;          ...
```

4 结语

本文简要介绍了信息系统中常用的公式定义方法,并对预定义与自定义相结合的方法进行了深入分析,最后以代理人佣金管理系统中核算佣金项为例,对相关佣金项的核算流程,自定义接口、自定义规则及相应的公式解析规则作了简要分析,并且给出了PL/SQL解析计算公式的部分代码。本文提到的方法简单实用,在处理只包含简单的运算符自定义公式时,有很好的借鉴作用,但在包含有复杂运算符的公式中,还必须考虑复杂计算符的正确解析,计算项之间的计

参考文献

- 1 RFC2460.Interneprotocol.version6(IPv6).1998.86-95.
- 2 Stevens Richard W. TCP/IP 详解(卷1:协议).北京:机械工业出版社,2000.95-115.
- 3 Wtight GR, Richard W. TCP/IP 详解(卷2:实现).陆雪莹,蒋慧译.北京:机械工业出版社,2000.50-60.
- 4 宋树彬,王能.无线传感网络上的超轻量化的 IPV6 协议栈.计算机应用,2007,27.
- 5 徐梅.基于 ARM 的精简 IPv6 协议栈的研究与设计.上海:华东师范大学,2008.
- 6 王丽.基于 μ C/OS-II 的嵌入式 TCP/IP 协议栈的研究[学位论文],2006.
- 7 何轩,夏应清.基于 μ C/OS-II 的嵌入式系统以太网通信功能的实现.自动化技术与应用,2008,12(15).
- 8 胡德斌.基于 μ C/OS-II 操作系统的嵌入式网络服务器的设计与实现.微计算机信息,2007,21(23).

算优先级等其它相关问题。

参考文献

- 1 陈国华.MIS 中自定义计算公式方法的研究与应用.计算机工程与设计,2008,29(15):3924-3925.
- 2 杨选辉.信息系统分析与设计.北京:清华大学出版社,2007.
- 3 冯敬峰.代理商佣金系统因子公式的实现方法.邮件设计技术,2007,(10):61.
- 4 陈承欢.管理信息系统开发案例教程.北京:人民邮电出版社,2009.
- 5 Feuerstein S. Oracle PL/SQL 最佳实践.北京:机械工业出版社,2009.
- 6 杨忠民,蒋新民,晁阳.Oracle 10g SQL 和 PL/SQL 编程指南.北京:清华大学出版社,2009.