

文章编号:1001-9081(2008)02-0406-03

一种嵌入式 IPv4/IPv6 双协议栈的实现

林 晓,雷洽军

(洛阳师范学院 信息技术学院,河南 洛阳 471022)

(linxiao6008@126.com)

摘 要:提出了一种基于平台无关性的嵌入式 IPv4/IPv6 双协议栈的实现方法。根据嵌入式应用,设计了相应的缓冲区管理机制,简化了 IPv6 模块中 ICMPv6 和邻居发现的相关功能以及 TCP 模块中的运行状态机和滑动窗口机制,并对协议栈接口层和传输层作了相应的功能扩展。

关键词:双协议栈;IPv6;TCP

中图分类号: TP393.04 **文献标志码:** A

Implementation of embedded IPv4/IPv6 dual-protocols stack

LIN Xiao, LEI Zhi-jun

(College of Information Technology, Luoyang Normal University, Luoyang Henan 471022, China)

Abstract: An implementation of embedded IPv4/IPv6 dual-protocols stack, which was independent of any platform, was proposed. According to the embedded application, the buffer management of the stack was designed. Some functions of ICMPv6 and neighbor discovery protocol were eliminated. And the process-state and glide window mechanism of TCP protocol were optimized and some functions of network interface layer and transport layer were extended.

Key words: dual-protocols stack; IPv6; TCP

0 引言

随着嵌入式系统与网络的日益结合,在嵌入式系统中引入 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。随着 Internet 技术的迅猛发展和规模的不断扩大,现行的 Internet 协议 IPv4 在地址空间、端到端的 IP 连接、服务质量和网络安全等方面暴露出了不足,极大地限制了 IP 网络的进一步发展。而 IPv6 所提供的巨大地址空间和网络安全等因素已经使其成为下一代因特网 IP 协议。为了能够支持 IPv6 的应用,并且考虑到从 IPv4 到 IPv6 仍然需要一个过渡期,本文实现了一种基于实时操作系统的 IPv4/IPv6 双协议栈结构。

1 协议栈的设计要点

1.1 平台无关性

TCP/IP 协议栈的进程模型有以下几种方式:

1) 每一个协议作为一个单独进程,要求严格分层。优点是协议层次清晰,代码易于调试和理解。缺点是需要同上下文切换,增加了系统开销。

2) 协议栈实现于操作系统内核当中。优点是协议栈与用户程序分离,保证了安全性。缺点是协议栈只能限定于特定的操作系统内核。

3) 所有的 TCP/IP 协议都驻留在一个进程当中,与操作系统内核分离。操作系统为其提供任务调度、信号量传递和进程同步等服务。

本文所设计的 TCP/IP 双协议栈,要求不能局限于特定的

硬件环境和软件系统。因此,本协议栈的实现采用第三种进程模型,协议栈模块与操作系统内核分开。通过定义一个操作系统模拟层实现平台无关性。操作系统模拟层屏蔽硬件上的差异和获取操作系统提供统一的接口,如时钟、过程同步以及信息传递机制等。对于特定的嵌入式操作系统,直接对操作系统模拟层相应部分进行修改,以符合具体的要求。操作系统模拟层主要包括下面两个部分:

1) 与处理器和编译器有关

对不同微处理器的 BYTE_ORDER 存储系统的定义;对不同编译器的数据类型长度的定义;与 CPU 或编译器相关的内容的定义。

2) 与操作系统相关

协议栈进程与应用程序进程之间的同步与通信,需要对信号量 sys_sem_t、消息队列 sys_mbox_t 和时钟函数进行定义和操作。

1.2 协议栈的裁减

标准的 TCP/IP 协议栈功能复杂,代码量大,且需大量的存储器资源。为了满足嵌入式应用,必须对 RFC 定义的协议栈进行裁减^[1]。本协议栈的实现作了如下裁减:

1) IP 模块。不支持数据分片,不支持数据包转发,不支持 IPv6 移动功能,不支持部分 IPv6 的扩展报头。

2) ICMPv6 模块。支持信息报文中的请求和应答报文,支持错误报文中的目的不可达和超时报文。

3) 邻居发现模块。支持地址解析、重复性探测,省略了路由功能。

4) TCP 模块。支持多个连接,简化滑动窗口,实现快速重

收稿日期:2007-09-10;修回日期:2007-11-20。 基金项目:国家自然科学基金资助项目(60475021)。

作者简介:林晓(1978-),女,河南南阳人,讲师,主要研究方向:计算机网络、嵌入式系统设计、人工智能;雷洽军(1960-),河南南阳人,男,副教授,主要研究方向:计算机网络、数据库。

传、慢启动与拥塞避免算法的 TCP 可靠性机制;约减掉半关闭、紧急指针、大部分 TCP 选项等。

协议栈的层次结构如图 1 所示。采用独立于网络层的方式,IPv4 协议和 IPv6 协议分开实现,条理清楚,易于裁减,同时不会增加太多代码空间。

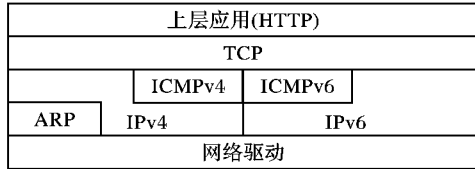


图 1 协议栈层次结构

2 协议栈的实现

2.1 内存管理

协议栈常用的节约内存的方法是零拷贝和一次拷贝技术。这两种方法的目的都是避免数据的频繁拷贝,提高协议栈运行效率。系统的实现采用了一次拷贝技术,应用程序和协议栈缓冲区分开,将应用程序数据拷贝到协议栈缓冲区,在协议栈内部使用零拷贝。

内存管理主要是针对协议栈内部缓冲区的管理。为了胜任任意大小报文的处理,并且能够达到比较高的处理效率,在这里定义了两种缓冲区实现机制:固定长度缓冲区(pbuf_pool)和变长度缓冲区(pbuf_ram),如图 2 所示。pbuf_pool 主要适合于网络设备驱动,存放从网卡接收的数据。pbuf_ram 根据数据的大小动态分配缓冲区,如协议的报头空间。

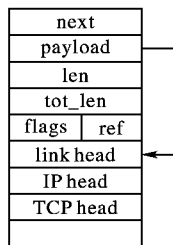


图 2 pbuf_ram 结构示意图

在这里对存储空间的应用做了一定改进。由于发送数据时很难确定各层协议头部需要多大的内存空间,特别是 IPv6 报文可以包含若干个扩展头,所以只申请基本头部的内存空间,在使用中动态调整的可能性就会很大,使得系统的效率降低。在这里提出了一种优化方法,根据局部性原理,通常情况下下一个连接在局部时间范围内需要网络层提供相同功能的概率非常大,所以在该连接的 TCP_PCB 控制块中增加一个变量 num,用于记录上次通信时 IPv6 协议头所使用的字节数,并且此变量可以被网络层的函数修改。当传输层下次申请 pbuf 时,就参考这个值。如果所需的存储空间比此值大,就需重新申请能满足要求的存储空间,同时修改 num 的值,并作为下一次申请的参考。

2.2 网络接口的实现

网络接口层在设计上主要考虑下面几个目标:1)向协议层提供一个统一的网络通信接口,能够将上层协议模块与具体的网络接口硬件隔离开来,以利于向不同操作系统的移植;2)支持一个网络接口配置多个 IP 地址的功能,因为 IPv6 一

个接口的 IP 地址包括本地链路、本地站点和全局三类地址;3)支持多播报文的收发,提供动态的多播 IP 与多播 MAC 地址的映射服务;4)提供较好的通信性能等。

针对上述目标,设计了如下的网络接口控制结构:

```

struct netif {
    struct netif * next;           /* 指向下一个 netif 结构的指针 */
    netif_ipaddr * iplist;        /* 指向接口的 ip 地址链的首指针 */
    char name[ 2];                /* 接口名称 */
    err_t ( * input)(struct pbuf * p, struct netif * );
    /* 接口输入函数指针 */
    err_t ( * output)(struct netif * , struct pbuf * p, netif_addr * );
    /* 接口输出函数指针 */
    err_t ( * linkoutput)(struct netif * , struct pbuf * p);
    /* 链路层输出函数指针 */
    void * state;                 /* 指向接口的状态信息 */
    mac_addr netifmacaddr;       /* 接口的 MAC 地址 */
    netif_multi * multilist;
    /* 本接口的 MAC 多播地址链首指针 */
    :                             /* 省略了部分信息 */
}

```

网络接口层是协议栈与下层网络设备驱动程序之间的接口,同时支持 IPv4 报文和 IPv6 报文,其功能扩展如下:

1)发送数据包处理。根据数据包的版本信息封装以太网报头,若为 IPv6 报文,不再需要地址解析。

2)接收数据包处理。将网络驱动程序接收到的数据分为三种:IPv4 报文、IPv6 报文和 ARP 报文,根据帧类型进行不同的处理。

2.3 IPv6 模块

IPv6^[2] 模块主要包括两部分的功能:1)负责从网络接口层接收 IPv6 数据包,对数据包进行一定的处理后发送给上层处理模块(TCP 或 ICMPv6);2)从上层接收数据,进行报文的选路,封装 IPv6 报头后将数据报发送给网络接口层。

2.3.1 ICMPv6^[3] 的简化分析

1)ICMPv6 信息报文

嵌入式设备只需作为 Ping 的接收端,所以不需要发送回显请求报文和处理回显应答报文。为了让客户端可以测试网络状态以及测试使用,必须支持接收请求报文和发送相应的应答报文。

2)ICMPv6 错误报文

由于嵌入式设备不会作为路由转发设备,所以需要实现发送端口不可达报文,同时要支持超时报文通告对端超时错误。在 BSD 中,IP 层没有发送参数问题报文,ICMPv6 无需支持处理参数问题报文。

2.3.2 邻居发现协议(NDP)的简化分析

NDP^[4] 是 IPv6 协议的一个基本组成部分,它用于地址解析、邻居发现以及路由器及网络参数发现。本协议栈不支持路由功能,所以无需实现发送路由器通告报文,但必须接受路由器通告报文,以完成路由发现功能。同时支持邻居通告和邻居请求报文的接收和发送,以实现地址解析和重复探测。

2.4 TCP 模块

在 TCP 之上实现有限的 Web 服务,可以对该协议的实现机制作相应的简化及优化。

2.4.1 TCP 模块的扩展

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

1) 数据结构扩展。

PCB 是用来记录每个 TCP 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,同时添加标识位区别连接。

2) 操作扩展。

在 TCP 处理过程中包括输入时对 IP 地址和端口号的匹配,输出时校验和的计算和路由的查找。由于对 IPv4 和 IPv6 报文调用相同的处理函数,需要在实现时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。

2.4.2 简化 TCP 状态机

TCP 的面向连接和可靠性传输等特性决定了每一个标准的 TCP 状态机庞大而且复杂,对于嵌入式应用,可以根据需要进行简化。

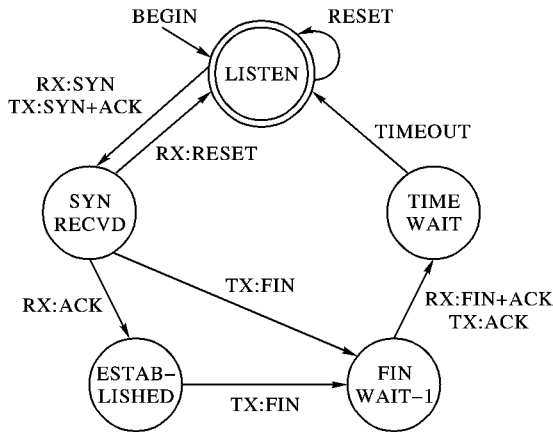


图 3 简化 TCP 有限状态机

建立连接分为主动打开和被动打开,本协议栈设计作为监听状态的服务器端,仅实现被动打开。在设计中去掉了 CLOSED 状态,让它一开始就处于 LISTEN 状态,等待对方发起连接。

断开连接分为主动断开和被动断开,从安全性的角度考虑,本设计仅实现主动断开连接。当服务器发送完数据时,向客户机发送关闭连接请求。

通过以上分析,根据特定的应用不失一般性地裁减掉复杂和冗余的机制,除去标准状态机中的客户端部分,简化连接的建立和关闭过程,就得到如图 3 所示的简化的 TCP 状态机模型。

2.4.3 简化的窗口机制

标准的 TCP 协议使用慢启动的滑动窗口机制^[5],它允许发送方在等待一个确认之前发送多个窗口。其确认是一种批量的确认,处理器要对多个数据报连续传输进行维护和处理,困难较大。

而滑动窗口的一个极限情况,是只使用单个窗口,这就变成了一种简单确认的处理方法。只是对于单个数据报的发送和确认,虽然节约了系统的资源,也使维护更加方便,但是却降低了效率。

系统采用了一个折中的办法,适当调整窗口的大小,限制连续发送报文段的个数,实现了一个简化的窗口机制,如图 4 所示。在这里,设置窗口大小 (snd_wnd) 为 4 个最大报文段长度 (MSS),可以适当提高数据传输效率。

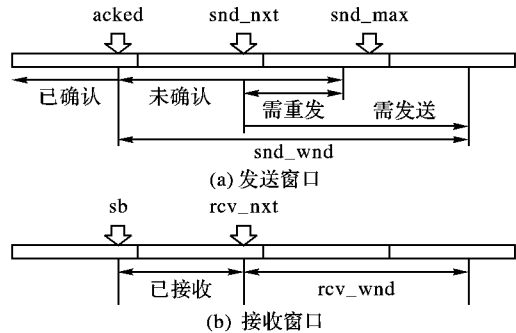


图 4 滑动窗口示意图

同时,避免了所谓的 TCP 糊涂窗口问题, TCP 报文段在建立和排队时不通告接收器窗口大小,如果 TCP 接收方通告了一个小窗口,就一直等待直至窗口有足够大的空间。

3 协议栈的移植和测试

在完成双协议栈的设计调试后,将此协议栈移植到基于 ARM7TDMI 核的 AT91M40800 开发板上,使用其中内嵌的 10 Mbps 以太网控制器 RTL8019AS 作为网络接口。在该平台上实现了一个简单的 Web Server。通过构建提供 IPv4 地址和 IPv6 地址访问的网络拓扑结构进行测试,该双协议栈运行稳定、正常。

同时,在相同环境下将此协议栈与开源的协议栈 LwIP 作了吞吐率性能方面的比较,如图 5 所示。可知,本协议栈有较好的吞吐率,在连接数达到 3 条以后,平均吞吐率达到了 460 kBps。

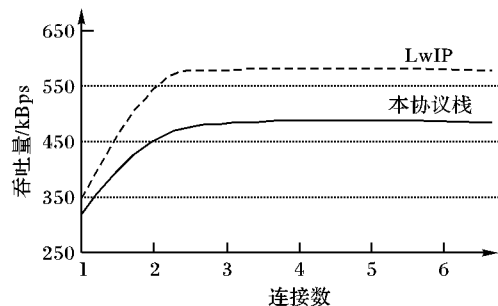


图 5 协议栈吞吐率比较

4 结语

本文主要介绍了一种嵌入式双协议栈的实现,实践证明它是一个符合 TCP/IP 协议规范的、高性能、低开销、可移植的 IPv4/IPv6 双协议栈。

参考文献:

- [1] OKABEN. Minimum Requirement of IPv6 for Low Cost Network Appliance[EB/OL]. [2007 - 03 - 12]. <http://www.ietf.org/internet-drafts>.
- [2] HAGENS. IPv6 精髓[M]. 技桥,译,北京:清华大学出版社,2004.
- [3] RFC 2463, Internet control message protocol (ICMPv6)[S], 1998.
- [4] RFC 2461, Neighbor discovery for IP version 6 (IPv6)[S], 1998.
- [5] STEVENSWR. 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, 1999: 209 - 224.