文章编号:1000-5013(2015)02-0179-06 **doi**:10.11830/ISSN.1000-5013.2015.02.0179

开源 Linux 的嵌入式安全 SOHO 路由器设计

朱龙,刘长君

(四川信息职业技术学院 图书馆,四川 广元 628040)

摘要: 在嵌入式平台上实现一个基于开源 Linux 系统,面向小规模网络的集成路由和防护功能的网络接入设备,并对其路由和网络防护性能进行测试.结果表明:系统可以满足小规模网络的接入和保护需求,支持直观简单的 Web 管理(WBM)配置方式,同时也为具备一定网络技能的用户提供传统的 Telnet 路由配置方式.

关键词: 网络防护;网络接入;嵌入式;路由器;Web 管理;Linux 系统

中图分类号: TP 393.08 **文献标志码**: A

随着互联网的迅速发展,SOHO(小型办公和家庭办公)用户数量迅速增加.为了向 SOHO 用户提供低成本、高性能的网络接入和网络防护,小型路由器和防火墙产品应运而生[1].小型路由器和防火墙大多是在嵌入式平台上运行比较成熟的商业软件系统,如 Router OS,Mono Wall 等.但使用商业软件系统不可避免地会引入较高的软件成本和版权问题.嵌入式系统发展到现在,其处理能力已经较为强大.在此基础上,本文在嵌入式平台上,设计一款同时具备路由功能和防火墙功能的网络接入设备,并对其网络防护性能进行测试.

1 系统平台分析

1.1 硬件平台与交叉编译

设计选用的硬件平台为华恒公司的 PPC860 嵌入式开发板,包含核心板和基板.核心板搭载了 MPC860 处理器,16 MB 的 SDRAM 及 4 MB 的 FLASH 存储器;基板为用户提供了外设接口,有 10 MB 以太网接口、100 MB 快速以太网接口,以及串口和用于调试的 BDM 口.

嵌入式开发过程中常用的开发方式是交叉编译,即在宿主机上编写程序,用交叉编译、汇编、链接工具形成可执行程序,然后在目标板上通过串口和以太网口以 mount 的方式下载宿主机中的可执行程序并调试、运行[2].在本设计中,宿主机中所使用的编译工具,是开发板自带开发套件中用于 Power PC 处理器指令集的 powerpc-gcc 编译器.在宿主机中用 powerpc-gcc 编译器编译、链接源程序,生成可用于开发板的可执行程序,供开发板通过交叉编译环境下载调试、运行.交叉编译环境的搭建过程如下:

1) 配置宿主机 NFS(Network File System,网络文件系统)和 TFTP 服务器;

2) 通过交换机建立宿主机到目标板的网络连接(也可以用交叉线直连宿主机与目标板);

3) 建立宿主机到目标板的串口连接;

4) 通过 minicom 程序经串口控制目标板,使其通过以太网口下载宿主机可执行程序,调试、运行.

1.2 系统功能模块分析

在设计的操作系统平台选择上,选用了 2.4 版本的 Linux 内核.虽然 2.6 版本内核已经比较成熟,但其抢占内核和崩溃恢复机制对本设计性能提高不大,因此选择比较稳定、驱动支持较好的 2.4 版本 Linux 内核.在 Linux 内核之上的软件架构按照功能划分为路由模块和防火墙功能模块.

收稿日期: 2015-03-20

通信作者: 朱龙(1975-),男,副教授,主要从事计算机网络、嵌入式系统开发的研究.E-mail:zhulong-1975@126.com.

基金项目: 国家社科基金重大项目(12&ZD003)

路由模块采用的是 Zebra-0.95a 开源路由软件,同时为用户提供灵活的 Telnet 配置方式;防火墙功能模块则是采用了 Linux 系统中常见的 Netfilter/Iptables 防火墙架构,通过 Iptables 工具配置防火墙规则,然后由 Linux 内核中的 Netfilter 机制实现规则的应用.在系统软件架构的最高层提供了用户基于 Web 管理的方式[3-4].

通过页面服务器 Boa 向用户提供 Web 的配置界面,以便对防火墙规则和简单路由进行配置,配置的实现则是由 Boa 执行后台 CGI 程序完成;Linux 内核及相应的功能模块存储在只读的 Cramfs 文件系统中;用户信息及路由、安全配置文件保存在读写的 JFFS2 文件系统中[5-6].系统软件功能模块分析,如图 1 所示.

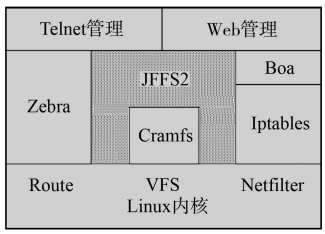

图 1 系统功能模块分析图

Fig. 1 System function module diagram

2 系统软件的架构设计

2.1 Linux 内核剪裁

使用开源 Linux 内核作为操作系统,以避免采用商业软件引入的软件成本和版权问题.为避免出现软件漏洞或安全问题,同时也出于对嵌入式硬件平台 Flash 存储器容量考虑,需要将 2.4.18 版本的 Linux 内核进行剪裁,以满足系统需求.

在内核源码路径下运行 make menuconfig 命令可对 Linux 内核进行剪裁.在剪裁过程中,保留了 Power PC 处理器支持、内存管理、进程管理、系统中断向量、文件系统支持等基本功能模块,以及以太网口驱动、串口驱动、能源管理等扩展功能模块.

在内核剪裁过程中,需要保证剪裁的内核对 ramdisk、JFF2 文件系统的支持[7].因为剪裁过的 Linux 内核以 ramdisk 系统烧写到开发板 Flash 存储器中,对 JFFS2 文件系统的支持则是为了实现大容量 Flash 存储器可读写文件系统扩展.在内核剪裁过程中要保留 Netfilter 功能模块;Netfilter 功能模块和 Iptables 防火墙配置软件共同实现设计的防火墙功能[8].

2.2 Zebra 路由模块设计

Zebra 是一个支持 RIP,OSPF 等路由协议,基于 TCP/IP 服务的开源路由软件.Zebra 支持 IPv4 和 IPv6,在实际应用中,具有较强的可扩展性.Zebra 提供了一个高质量的多服务路由引擎,它为每个路由协议提供交互接口,并支持通用客户端命令.在设计中采用了 Zebra0.95a 作为路由功能模块.

Zebra 的源码包中指定的用于编译、重建的编译器默认为适用于 X86 架构的 gcc 编译器.在本设计中,要将 Zebra 路由软件运行于 PowerPC 架构的嵌入式开发平台上,因此在配置源码之前,要对配置文件进行修改.在 Zebra 源码路径下用 CC=/LinuxPPC/CDK/bin/PowerPC-gcc 命令来指定配置源码所使用的编译器.其中,/LinuxPPC 路径为开发板所使用开发套件在宿主机中的安装路径,/LinuxPPC/CDK/bin/PowerPC-gcc 指定了用于 PowerPC 架构处理器的 C 语言编译器.

执行 Zebra 源码路径下的 configure 配置脚本,并对配置过的源码用 make 命令进行编译,最后用 make install 命令完成安装.Zebra 安装完成之后还要对其配置文件/usr/local/etc/zebra.conf 进行修改,使之符合系统需求,同时修改相同路径下的 RIP,OSPF,BGP 路由协议的配置文件.完成配置之后,以后台守护进程的方式执行 Zebra 路由软件——zebra-d.这样 Zebra 路由软件就在开发板上运行,宿主机或者同一网段内其他主机可以通过 Telnet 方式访问并对其进行配置.Zebra 路由模块正常运行时,对其远程 Telnet 登录将出现用户登录密码输入提示.

2.3 Netfilter/Iptables 防火墙模块设计

在 1.1 版本的 Linux 内核中提供了基本的包过滤功能,在经历过 2.2 版本 Linux 中的 Ipchains 防火墙机制之后,当前在 2.4 版本 Linux 内核中使用的是 Netfilter/Iptables 机制的第三代包过滤防火墙.其中,Netfilter 组件运行于内核空间,是 Linux 内核的一部分,这也是在进行 Linux 内核剪裁过程中,要求保留 Netfilter 功能模块的原因.它是由一些信息过滤表组成,包含了内核用来控制数据包过滤处理的规则集.Iptables 组件是一个防火墙规则配置工具,运行于用户空间,Iptables 向用户提供一个命令行模式的防火墙规则管理工具,使用户可以灵活准确的插入、修改、删除数据包过滤表中的防火墙规则.

Iptables 的板上移植过程与 Zebra 类似,首先通过指定开发板所用编译器对源码进行配置、编译和

重建，然后将生成的可执行程序下载到开发板运行.

在防火墙功能模块设计过程中，着重关注两个要点：一是防火墙功能扩展；二是对防火墙规则冲突的检测和避免.

2.3.1　*Iptables 功能模块扩展*　Iptables 提供了一种灵活准确的防火墙规则配置方式，它支持部分基本的配置规则，例如数据包协议匹配、端口匹配和 IP 地址匹配等. 但是在实际应用中对防火墙的配置需求更加复杂，例如字符串匹配的内容过滤、连接数量匹配、防护策略时间规则和 P2P 数据包匹配等. 这些复杂的防火墙规则实现是通过安装特定的 Netfilter/Iptables 功能 patch 实现的. 在本设计中使用的 4 种扩展规则模块 .

1) Iptables string match. 这个 patch 中增加了 CONFIG_IP_NF_MATCH_STRING，允许在一个数据包里匹配一个字符串，用来进行内容过滤.

2) Iptables connlimit match. 这个 patch 为 Iptbales 扩展了连接数量匹配，用于限制每个远程 IP 地址的并发 TCP 连接数量.

3) Detects some P2P packets. 这个 patch 可以匹配部分 P2P 数据包，用于控制网络流量. 这个 patch 结合连接跟踪，与数据包调度器配合，用于 P2P 流量计算和整形.

4) Iptables time match：这个 patch 为 Iptables 增加时间匹配功能，允许对其他防护规则添加时间匹配选项，可以使规则在特定时间范围内启动或关闭.

除了上述扩展功能模块，用户还可以通过指定协议类型、IP 地址、端口号来自定义防护规则，设计的防火墙功能模块可以满足日常网络防护的需求.

2.3.2　*防火墙规则冲突检测避免*　Iptables 提供防火墙规则配置，但是它只是顺序执行已有的防火墙规则，而不对规则冲突进行检测. Iptables 只对防火墙规则的语法格式进行检查，如果防火墙规则前后对某一个数据包配置了接受和拒绝两种策略，那么它将执行先配置的策略，而不会提示规则冲突. 为了避免出现这种配置过程中规则的前后冲突，在设计中使用了基于文本的规则冲突检测避免机制[9-11].

在本设计中，防火墙基本设置是通过 Web 方式来配置的，页面中采用单选框的形式保证了规则的一致性. 因此，防火墙规则冲突主要存在于用户自定义规则集中. 用户自定义规则主要包括六个字段：协议类型、源 IP 地址、源端口号、目的 IP 地址、目的端口号、策略.

规则冲突检测避免机制工作原理：接收 Iptables 的配置命令，写入防火墙规则脚本 iptable. rules 中，将自定义的规则字段存储到 rule 结构体中. rule 结构体定义如下所示.

```
struct rule {
    bool protocol; //协议类型，识别 TCP 或 UDP
    int src_ip[4]; //源 IP 地址
    int src_port; //源端口号
    int dst_ip[4]; //目的 IP 地址
    int dst_port; //目的端口号
    bool policy; //匹配策略，接受或丢弃
    struct rule * next; //指向下一条规则
}
```

所有规则的 rule 以链表的形式存储. 添加新自定义规则时，先逐项匹配链表中已有的规则，如果前 5 个字段一致，而 policy 字段不同，则说明新定义规则与已有规则冲突，需要修改；如果 6 个字段均一致，则说明新定义规则之前已经定义，属于冗余，可以删除；如果前面两种情况均不满足，则说明此规则为新规则，将此规则添加到规则脚本中.

冲突检测功能代码示意如下：

```
while(p_rule ! = NULL)
{
    if(p_rule. protocol == n_rule. protocol && \
    p_rule. src_ip == n_rule. src_ip && \
```

```
p_rule.src_port == n_rule.src_port && \
p_rule.dst_ip == n_rule.dst_ip && \
p_rule.dst_port == n_rule.dst_port )
    {
        if(p_rule.policy == n_rule.policy)
            {
                rule_redundancy();
                break;
            }
        else
            {
                rule_collision();
                break;
                }
            }
        p_rule = p_rule.next;
    }
if (p_rule = = NULL)
        rule_add();
```

其中:p_rule为指向已有规则链表的rule结构体类型指针;n_rule为用户提交添加的nrule结构体类型变量;nrule与rule结构体类型定义区别在于nrule中没有形成链表的next指针.通过这种基于文本的规则冲突检测、避免机制避免防火墙规则集的二义性和冗余,为用户提供准确添加自定义规则的环境.

2.4 Web配置模块设计

2.4.1 *Boa页面服务器简介* Zebra向用户提供了基于Telnet的远程配置方式,但是这种配置方式采用的是通用命令行,需要一定的网络技术基础.为了便于普通用户对设备进行直观简单的配置,本设计为用户提供了基于Web的页面配置方式.

本设计向用户提供了基于Web的配置方式,这首先需要一个Web页面服务器.出于硬件平台性能考虑,选用了再嵌入式开发中常用的页面服务器Boa.它是一个单任务的HTTP服务器软件,只能依次完成用户的请求而不会fork出新的进程来处理并发的连接请求,但是可以fork出新进程供CGI(Common Gateway Interface,通用网关接口)程序运行;由于设备配置过程不需要并发机制,因此Boa页面服务器符合设计需求[12].Boa页面服务器的移植过程较简单,可参考Zebra移植过程.

2.4.2 *交互配置流程实现* Boa向用户提供前台HTML配置页面,同时负责接收用户配置请求,以环境变量的方式传递给后台CGI程序,由CGI程序解释配置请求,并做出相应的配置动作[13].配置有如下5个具体过程(图2).

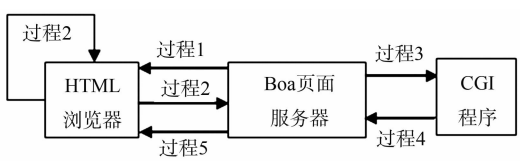

图2 交互配置过程示意图

Fig. 2 Schematic diagram of interactive configuration process

1) 表示Boa页面服务器生成Web配置页面,在远程配置端由HTML浏览器解释.

2) 用户向HTML页面中以表单形式填写配置信息,若配置信息不合法,则输出警报,要求重新填写;若配置信息合法,则将表单提交至Boa服务器.

3) Boa服务器将用户配置信息以环境变量的方式传递给后台CGI程序处理.CGI程序获取用户配置信息是通过读取CONTENT_LENGTH环境变量实现的,CGI程序以字符串形式获取用户提交配置表单内容,然后通过对字符串的处理来获取并解释用户配置信息.

4) CGI进行系统调用,完成用户请求配置后,以HTML代码的形式向Boa服务器返回处理结果.CGI对用户请求进行处理,得到相应配置命令,经系统调用完成对路由功能或防火墙功能的配置.其中,

在对防火墙功能配置过程中，系统调用之前要先调用前文所述防火墙规则冲突检测、避免机制[14].

5) Boa 服务器将 CGI 返回的处理结果传送至用户端 HTML 浏览器，向用户显示当次配置请求的处理结果.

2.4.3　*应用配置例析*　以对 ICMP 协议的操作为例，分析 Web 方式管理的工作过程. 用户通过浏览器向 Boa 服务器提交配置表单，在此处，表单内容很简单，包括两个单选框，Y 表示阻止 ICMP 协议，N 表示允许 ICMP 协议，缺省状况允许 ICMP 协议，用户配置点选 Y 项，即阻止 ICMP 协议. 即

```
〈form name=cfg method=post action=/cgi-bin/cfg〉
ICMP Forbidden
〈input type=radio name=yn value=0〉Y
〈input type=radio name=yn value=1 checked〉N
〈input type=submit name=submit value=submit〉
```

Boa 接收到表单提交信息后，调用表单 action 选项中指定的后台 CGI 程序 cfg 来处理该配置表单信息，Boa 以字符串形势将表单信息传递给 CGI 程序 cfg，cfg 读取环境变量，即

```
n=atoi(getenv("CONTENT_LENGTH"));
for(i=0;i<n;i++)
in[i]=getc(stdin)
```

此时，in 数组中的内容：yn=0&&submit=submit. cfg 读取 in[3]，与"0"字符进行比对，如果匹配，则通过 system 系统调用执行 Iptables 规则，iptables -A INPUT -p ICMP -j Drop；否则，执行 ACCEPT 策略，然后将此条规则添加到规则脚本文件中保存. 完成操作后，向 Boa 提交返回到用户浏览器的 HTML 页面，提示操作完成. WBM 管理页面，如图 3 所示.

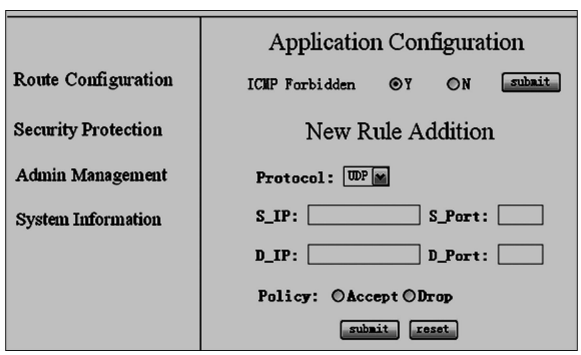

图 3　安全 SOHO 路由器 Web 管理页面

Fig. 3　Secure SOHO router Web management page

3　系统性能测试

3.1　路由配置模块测试

设计实现的路由功能模块有两种配置方式，可以通过 Web 管理(Web based management，WBM)配置方式进行配置，也可以通过 Telnet 方式进行配置. WBM 配置方式较为直观、简单，但配置不如 Telnet 后的命令行方式灵活. 为了对路由模块的功能有更全面的了解，本部分测试是 Telnet 登陆设备之后的结果，而 WBM 配置测试是与防火墙功能模块统一进行测试的.

Telnet 方式登陆设备之后，提供了类似思科路由的通用命令行配置界面：用户模式、特权模式、配置模式. 配置模式下的命令列表，如图 4 所示.

测试结果表明：路由模块可以正常工作，并且 Telnet 配置方式提供了类似思科路由器的通用命令行界面，配置方式灵活，功能完备.

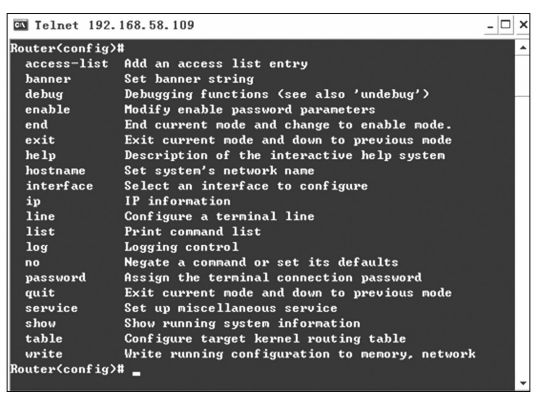

图 4　配置模式下的命令列表

Fig. 4　Configuration mode command list

3.2　防火墙基本规则测试

在对防火墙功能模块的测试过程中，以防止外部网络 Ping 入为例，对其基本防护规则进行了测试.

为了对外部网络屏蔽内网信息，防火墙基本规则中通常会拒绝外部网络使用 Ping 命令嗅探内部网络信息. 在本次测试中，在配置之前，外部计算机可以通过 Ping 命令嗅探到防火墙；在配置了丢弃 IC-

MP协议数据之后,外部计算机用Ping命令嗅探防火墙提示请求超时错误.测试表明:本设计的防火墙功能模块实现了基本的应用规则防护.

4 结束语

设计并实现一个基于开源Linux的,集成了路由功能和网络防护功能的嵌入式网络接入设备.在开发板上实现了路由功能模块和防火墙功能模块.然后,分别从路由配置模块、防火墙基本规则设置两个方面进行测试.结果表明:本设计可以为用户提供直接灵活的配置方式,实现了路由功能和安全防护功能在同一开发平台的集成,所得到的网络设备可以满足小规模网络的接入和防护需求.

参考文献:

[1] 游侃民,朱宁西.基于嵌入式Linux的SOHO路由器设计[J].微计算机信息,2009,25(11):19-31.
[2] 褚文奎,樊晓光,黄培成.基于PowerPC处理器MPC8250的嵌入式Linux系统开发[J].计算机工程与设计,2006,27(1):179-181.
[3] 彭涛,李声晋,芦刚,等.远程设备监控系统中嵌入式Web服务器的设计[J].机械与电子,2008(1):65-68.
[4] 姚晓宇,赵晨.Linux内核防火墙Netfilter实现与应用研究[J].计算机工程,2003,29(8):112-113,163.
[5] 王磊,谢维波.AdHoc网络在嵌入式Linux上的实现[J].华侨大学学报:自然科学版,2011,32(2):161-164.
[6] 张祖鹰.嵌入式Linux系统的网络实现究[J].船海工程,2009,38(增刊1):83-85.
[7] 刘文峰,李程远,李善平.嵌入式Linux操作系统的研究[J].浙江大学学报:工学版,2004,38(4):447-452.
[8] 陈闵中.Linux在嵌入式操作系统中的应用[J].同济大学学报:自然科学版, 2001,29(5):564-566.
[9] 张昭理,洪帆,肖海军.一种防火墙规则冲突检测算法[J].计算机工程与应用,2007,43(15):111-112.
[10] 赵波,秦涛,张新有.嵌入式防火墙规则冲突检测算法的实现[J].实验科学与技术,2009,7(6):153-156.
[11] 张国定,李随意,张翰林,等.基于IPv6的SOHO宽带路由器的软件设计[J].河南教育学院学报:自然科学版,2008,17(2):33-34.
[12] 张娟,张雪兰.基于嵌入式Linux的GUI应用程序的实现[J].计算机应用,2003,23(4):11-13.
[13] 王金东,赵海,韩光洁,等.基于SNMP的嵌入式系统Web管理模型 [J].计算机工程,2005,31(1):39-40,59.
[14] 郭松,谢维波.Linux下Proc文件系统的编程剖析[J].华侨大学学报:自然科学版,2010,31(5):515-520.

Design of an Embedded Security SOHO Router Apparatus of the Open Source Linux

ZHU Long, LIU Chang-jun

(Library, Sichuan Information Technology College, Guangyuan 628040, China)

Abstract: Based on the realization of a Linux open source system in the embedded platform, network access equipment for small scale network integrated routing and protection function, and test the performance of routing and network protection. The results show that the system can meet the access network and the protecting, supporting Web management simple and intuitive (WBM) configuration, but also provide the traditional Telnet routing configuration to the user who have a certain network skills.

Keywords: network protection; access network; embedded system; router; Web management; Linux system

(**责任编辑:** 黄晓楠 **英文审校:** 吴逢铁)