第32卷 第16期 计 算 机 工 程 2006年8月
Vol.32 № 16 Computer Engineering August 2006
·开发研究与设计技术· 文章编号：1000—3428(2006)16—0269—02 文献标识码：A 中图分类号：TP308

基于 Linux 的开放实验室管理平台设计

陈国震
（浙江纺织服装学院，宁波 315211）

摘 要：介绍了基于 Linux 的开放实验室管理优势，利用 Linux 强有力的网络性能、完善的软件支持，实现开放实验室管理，并给出了基于 liunx 的网络计费系统的设计方案。该文提出的方法是可行的、有效的，其效率大大高于一些起初的模型，且易于实现。
关键词：Linux；开放实验室管理优势；网络流量计费

Design of Management Platform for Computer Open Laboratory Based on Linux

CHEN Guozheng
(Zhejiang Textile and Fashion College, Ningbo 315211)

【Abstract】This paper introduces advantage of open computer laboratory management based on Linux, makes use of the powerful network in Linux function, perfect software supports, realizing to open the laboratory management. It puts forward the design solution of the network flow account based on Linux. It's proved that the design is available and feasible, the method is more efficient than former models and can be easily realized in practices.
【Key words】Linux; Advantage of open computer laboratory management; Network account

随着 Linux 应用的日益广泛，大量的网络服务器使用 Linux 操作系统，利用 Linux 作为开放实验室的网络支撑几乎已经成为事实。在 Linux 操作系统的基础上，构建开放实验室管理信息系统具有十分深远的现实意义。

1 基于 Linux 的开放实验室管理的优点

Linux 作为新兴的操作系统，在高校内首先得到普及，开放实验室管理中的服务器采用 Linux 操作系统的比例逐年增大。节约管理软件投入成本是显而易见的，Linux 稳定的性能和优秀的网络能力也是开放实验室管理的坚实基础。充分挖掘 Linux 的功能特性使之配合开放实验室管理的工作将是今后开放实验室管理的重要内容。基于 Linux 的开放实验室管理其优点如下：

(1)节省成本

首先在软件价格上，Linux 是基于开放源代码的操作系统产品，软件本身的价格较为低廉；其次在维护成本上，Linux 服务器在安装完之后便可长期稳定地工作，无需频繁重启、甚至于重装系统；再次在外网接入成本的节省上，Linux 服务器无论是使用代理服务或是路由接入，都不需要昂贵的百兆带宽接入，采用 10Mbps 的接入就可满足速度的要求，可充分利用带宽。

(2)安全性强

对于常见的 Windows 的病毒、木马程序对 Linux 系统丝毫不起作用，而针对 Linux 的病毒也相对较少，即使客户端感染了病毒，Linux 服务器也可以安然无恙，不用重装服务器系统。

2 开放实验室的网络流量计费软件的设计

实现网络计费通常有 3 种方式：(1)使用代理服务器方式；(2)使用路由器内部功能来实现计费；(3)网络监听计费。如图 1 所示。

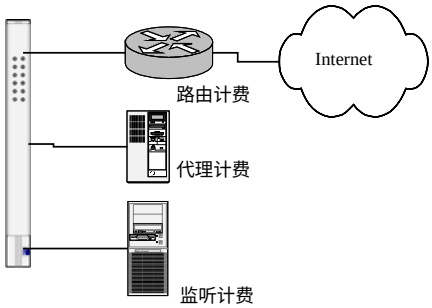

图 1 网络计费

(1)代理服务器计费：通过代理服务器的日志功能来进行计费。这种情况，可以在代理服务器上安装一个计费程序，从日志中读取数据进行计费。

(2)路由器计费：通过路由器本固有的功能来实现计费。路由器是内网和外网连接的通道，内网与外网之间的网络流量都必须经过路由器。所以是计费软件采集流量数据的地方，大部分路由器都具有流量记录的功能。计费系统把这些流量信息从路由器中读出来，经过程序处理生成按每个 IP 地址流量的计费账单。

(3)网络监听计费：通过监听网络上的数据，进行分析，存储。这种计费方式与路由器计费方式相同，只是把数据采集，分析，存储工作交给监听服务器来完成。以下着重讨论网络监听计费。

2.1 流量采集系统的分析和设计

2.1.1 基于 Linux 计费软件主要解决的问题

(1)获取流量：实现网络流量计费，必须先解决每台机器

作者简介：陈国震(1965—)，男，实验师，主研方向：计算机网络安全
收稿日期：2006-04-11 **E-mail：**chenuozheng@163.com

在任一时间段内的网络数据流量。得到上网计算机使用的网络流量最好的办法是通过获取该机器的发送/接收的 IP 包来进行。系统获取 IP 包是通过采用高效的流量捕捉程序来采集各个网段的网络流量，并按照各个计算机、流量的类型(HTTP 服务、Mail 服务、FTP 服务、Telnet 服务等)进行分类，按照设定的时间周期性地入库。然后对需要计费的网络流量，按照网络管理员设定的计费标准实现对流量的计费。

(2)账户管理：基于流量的网络计费软件还要求对流量按上网用户进行分类汇总记入用户的账户上。

通过防火墙过滤规则设定账户，对合法的用户给与放行，从而实现账户管理。不需要管理员干预，系统会自动管理已经登记的合法用户。客户端用户输入账号和口令，计费系统就可开始统计流量，当用户下机后，计费系统将该用户的网络流量记入该用户的账户上，同时记录用户的网络服务类型和时间等信息，并释放资源。管理员或用户在 Web 业务查询端，查询网络流量、计费情况等，因为这些处理所涉及的数据都在数据库中，可以通过 Web 服务器来实现。

2.1.2 流量计费采集系统的分析和设计

计费信息的采集包含 2 大部分 :数据的捕获及分析处理。

(1)数据的捕获

数据捕获模块使用Linux下的libpcap函数库来完成将数据捕获，然后返回该数据块的指针给数据分析模块[2]。

在 linux 下监听网络，应先设置网卡状态，使其处于杂混模式以便监听网络上的所有数据帧。通过 set_if_promisc 函数实现。然后捕获所有的数据帧。这样就可以捕获底层数据帧，返回的将是一个指向数据的指针，使用自定义函数分析数据帧。设置基本的数据帧头结构如下：

```
struct etherpacket
{struct ethhdr eth;
struct Iphdr Ip;
struct tcphdr tcp;
char buff[8192];
} ep;
struct ethhdr
{unsigned char desmac [6];
unsigned char srcmac [6];
unsigned short proto;}
```

其中，desmac [6]]是 48 位的目标地址的网卡物理地址，srcmac 是 48 位的源地址的物理网卡地址。proto是 16 位的以太网协议。其中主要有 0x0800 Ip，0x8035.x25,0x8137 Ipx,0x8863-0x8864 pppoe(这是linux 的 ppp),0x0600 ether_loop_back ,0x0200-0x0201 pup[1]等。

Ip 协议的报头结构如下：

```
struct   Iphdr{
_u8   tos;
_16   tot_len;
_u16   id;
_u16   frag_off;
_u8    ttl;
_u8    protocol;
_u16   check;
_u32   srcIp;
_u32   dstIp;};
```

这是linux 的Ip协议报头，protocol是Ip的协议分类主要有 0x06 tcp.0x11 udp,0x01 icmp,0x02 igmp等，srcIp 是 32 位的源Ip地址，dstIp是 32 位的目标Ip地址。其它协议结构在此不一一列出[3]。

利用Linux下libpcap函数库的调用来实现数据捕获[4]。

```
pcap_t *t;
char devicename [10];
int snaplen;
t=pcap_open_live(devicenname, snaplen, promisc, to_ms, errbuf);
```

本函数用于打开网卡用于捕获数据报，devicename 指定网络接口设备名，snaplen 指定单包最大捕捉字节数。promisc 指定网络接口是否进入混杂模式 to_ms 指定 ms 级读超时，如果调用失败返回 NULL，errbuf 包含失败原因。

```
int pcap_loop(t, int   cnt, treat,user);
```

本函数用于读取和处理数据报。其中 t 是 pcap_t* p:pcap_open_live 返回的数据报捕获的指针；int cnt:规定了函数返回前应处理的数据报数目；treat:为自定义的函数，在处理每个报后自动调用该函数进行再处理；user 为数据指针。

treat 函数，作为 pcap_loop 的 callback 函数，每当接收到一个数据帧，pcap_loop 调用 treat，并将接收到数据指针 user 传递 treat。

```
treat = lookup_treat(pcap_datalink(t));
```

pcap_datalink(t)获得 t 指向的数据帧的类型，自定义函数 lookup_treat 根据网络连接类型，选择相应的分析函数。

(2)分析处理

查出此包的协议类型(TCP/UDP/ICMP等)、源IP地址和目的IP地址以及源端口号和目的端口号[5]，按照头部信息中的数据长度作为该数据包的最终长度，按端口号和类型分类流量数据写入账户数据库中。

分析数据包类型的代码段如下：

```
if(FormM->swaps(EtherHead->FrameType)==ETHER_PROTOCOL_IP) //分析出以太网 IP 数据报文
{…if(t->protocol==6          //TCP(6){
…//协议值为 6，即是 TCP 的数据报文}
if(t->protocol==17)     /UDP(17)
{…  /协议值为 17，即是 UDP 的数据报文}
if(t>protocol==3) //ICMP 的数据报文
{…//协议值为 3，即是 ICMP 的数据报文}…
```

2.2 Web 设计

数据库表 1 结构包括如下内容：(1)时间；(2)源地址；(3)源端口；(4)目标地址；(5)目标端口；(6)协议；(7)包数量；(8)流量。

数据库表 2 结构包括如下内容：(1)源地址；(2)源端口；(3)目标地址；(4)目标端口；(5)协议；(6)包数量；(7)净流量；(8)发出流量；(9)接收流量；(10)捕获时间。

系统 Web 查询如表 1 所示。

表 1 计费查询

班级 04 模具　用户名 林华							
时间	源地址	源端口	目标地址	目标端口	协议	包数量	流量
16:21	220.181.28.42	4001	172.16.1.10	4000	UDP	1	210
16:21	220.181.28.42	4001	172.16.1.10	1660	UDP	1	290
16:21	220.181.28.42	4001	172.16.1.10	1665	UDP	2	350
16:21	61.153.17.56	21	172.16.1.10	1650	UDP	1	300

(下转第 279 页)