精通Linux丛书

Linux网络编程

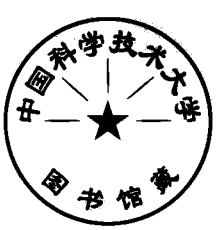
朱刚等

科学出版社

精通 Linux 丛书

Linux 网络编程

朱 刚 等 编著



科学出版社

2000

内 容 简 介

本书深入浅出地阐述了网络编程的基本思路以及网络编程中常用的客户机-服务器模型，详细地介绍了包括 TCP 套接口，UDP 套接口和原始套接口等套接口编程的各个细节，并且给出了大量例子。同时，本书对 CGI 的基本概念也进行了简单的介绍，并对 Perl 进行讲解。

本书可供广大的 Linux 爱好者，尤其 Linux 网络编程人员作网络编程的入门指导和参考用书。

图书在版编目（CIP）数据

Linux 网络编程/朱刚等编著. -北京：科学出版社，2000

（精通 Linux 丛书）

ISBN 7-03-007950-7

Ⅰ.L… Ⅱ.朱… Ⅲ.Linux 操作系统-系统管理-指南 Ⅳ.TP316.89

中国版本图书馆 CIP 数据核字（2000）第 61901 号

科学出版社 出版

北京东黄城根北街 16 号

邮政编码：100717

北京双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2000 年 8 月第 一 版 开本：787×1092 1/16

2000 年 8 月第一次印刷 印张：14 1/4

印数：1—5 000 字数：338 000

定价：19.00 元

(如有印装质量问题,我社负责调换<环伟>)

前　言

计算机自问世以来，操作系统作为计算机硬件和应用程序之间的纽带，它的更新一直备受瞩目，从DOS到Windows，每一个新的操作系统的问世，就必然带动着基于该操作系统软件的发展。目前个人计算机的操作系统基本上由Windows占领，但是大、中型的计算机系统，尤其是在网络及数据库方面的应用，都由UNIX等操作系统占领。随着网络化进程以及Internet的迅猛发展，即使对于个人来说，网络将成为获取信息的主要手段。

Linux的出现对于个人用户来说，无疑是一个福音，它是一个源代码公开、完全免费的UNIX操作系统，世界各地的自愿者们为这个充满前途的操作系统的发展贡献自己的才能。Linux操作系统作为UNIX的一个变种版本，继承了UNIX的诸多优点，如安全性、稳定性等。Linux已经为许多中小型的网络应用提供了完全的解决方案，充分显示了其优于Windows NT的网络功能。鉴于目前市场上关于Linux网络编程的书很少，我们编写了这本书供Linux网络编程人员的入门参考书，同时本书也是一本好的参考书。

本书深入浅出地阐述了网络编程的基本思路以及网络编程中常用的客户机-服务器模型，详细地介绍了套接口（TCP套接口、UDP套接口和原始套接口）编程的各个细节，并且给出了大量的例子，以加深理解。同时，本书对于CGI的基本概念也进行了简单的介绍，并对Perl进行了讲解。

本书由紫寒云创作室策划。参加编写的人员有朱刚、李奇志、傅宇旭、林依云、胡炎平、刘常青、罗斌、罗玲、段舸、周忠辉、刘丽辉、陈星田、冯江丽、张丽红、冯静、毛志刚、叶岳辉、肖志刚、林萍、任建国、文建军、易浩波、李晶、张伍等人。

由于编者水平有限，敬请各位读者和Linux爱好者给予批评和指正！同时对给予帮助的同事们表示最诚挚的感谢！

第 1 章　Linux 网络配置

众所周知：Linux 系统的安装与配置不像 Windows 系统那样简单，其网络配置更是需要许多专业知识，才能配置出高效的网络。为了在以后的章节中调试程序方便，先简单地介绍一下 Linux 的网络安装与配置。

1.1　Linux 内核对网络的支持

1.1.1　对协议的支持

在配置网络以前，首先必须知道网络使用的是什么协议，现在由于硬件发展得很快而不需要太多考虑协议栈的大小对网络的影响，尤其是在小型的局域网中更是如此，使用最通用、最流行的 TCP/IP 协议为网络基准协议的局域以太网已经相当普及。但是，还是有一些含有其他网络硬件的网络为提供支持使用了一些其他网络协议。

内核对网络协议的支持选项可以在编译内核的时候做出选择，图 1.1 是编译内核时关于网络协议支持选项的画面。

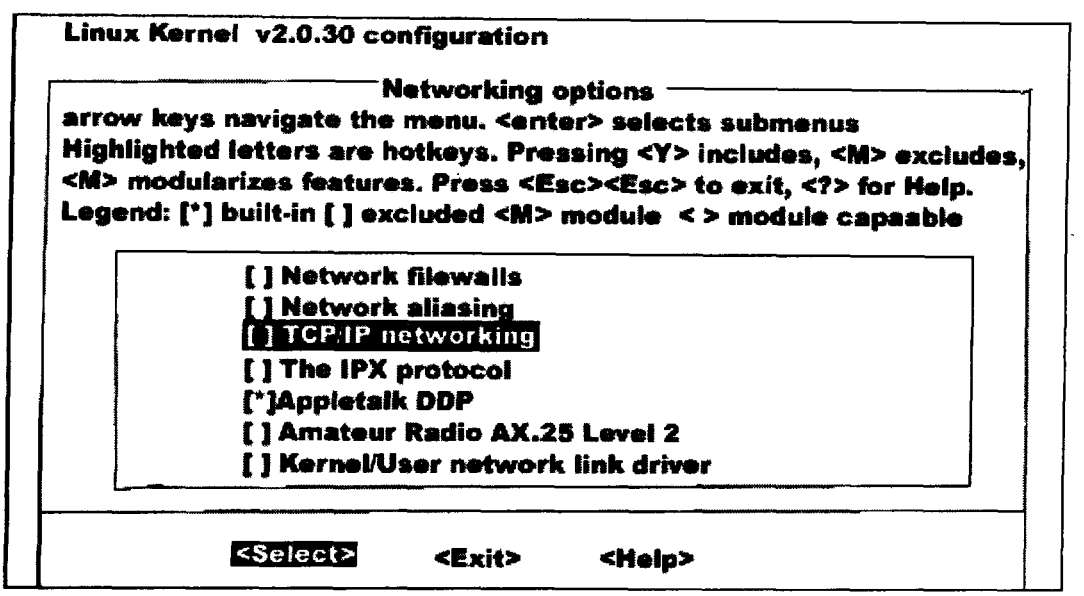


图 1.1　编译内核的网络协议选项

从图 1.1 中可以看出，内核主要支持的网络协议有：

- TCP/IP 协议以及在该协议下面的各种相关选项。
- IPX 协议，该协议是由 Novell 公司随其拳头产品 Netware 一起推出的，所以该协

议其实现在主要就是在使用Netware 服务器或者是客户机的网络里面使用。该协议之所以曾经很流行是因为IPX 协议是一个比较成功的支持路由的协议，而且该协议的整个协议栈的大小比较小，对网络速度的提高比较有利。

- Appletalk 协议，顾名思义，该协议是使用于与 Apple 机通信时使用的苹果公司的网络协议。
- Amateur Radio AX. 25 是业余无线电网络使用的通信标准，可以使用无线方式来建立网络连接。

1.1.2 对网络设备的支持

同样，使用在编译内核时候的选项来说明内核对网络设备的支持情况，如图 1.2 所示。

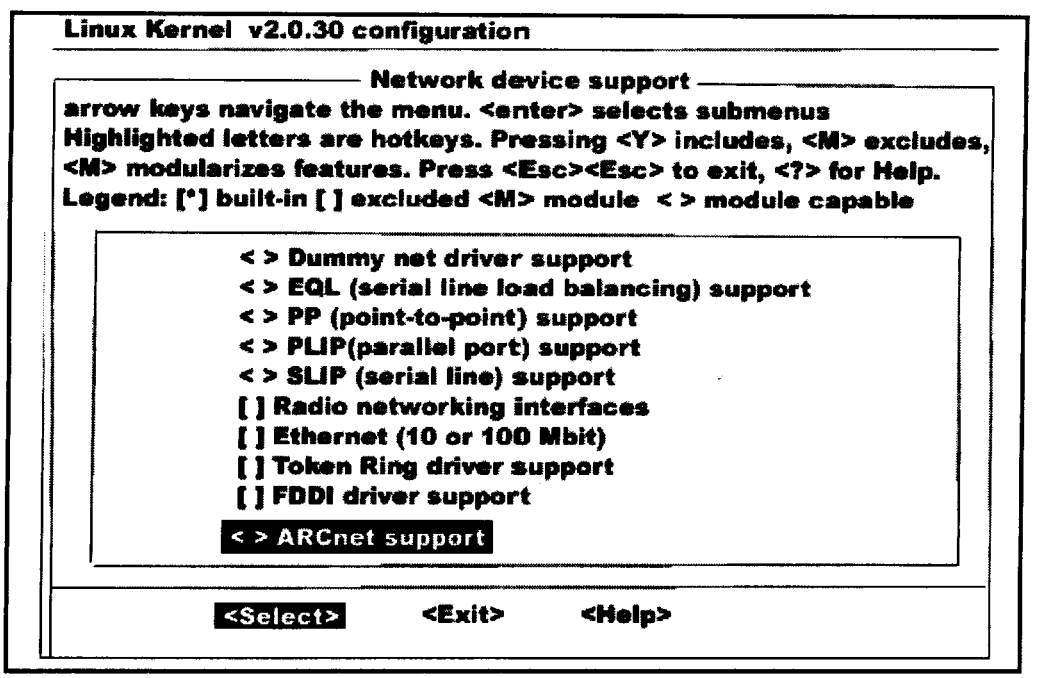


图 1.2 编译内核的网络设备选项

从图 1.2 中可以看出内核支持的网络设备有：

- Dummy net driver 的支持，该设备其实是一个虚拟的设备，其主要目的是当你在使用SLIP (Serial Line Internet Protocol，串行线 Internet 协议)、PPP (Point to Point Protocol 点对点协议，监控数据在电话线上传输正确) 这样的拨号或其他方式的连接时，通过该设备后对于本地应用程序来说，由服务器动态分配的地址看上去就像你有固定的、可以自己配置的 IP 地址。
- PLIP 设备的支持，PLIP 其实是Parallel Line Internet Protocol 的缩写，也就是并口 Internet 协议的网络设备。其实在 Linux 下面的 PLIP 和 SLIP 是有所不同的，因为 SLIP 多半通过电话线连接，而并口一般是直接的连接，所以该网络设备比较适合于没有网络接口卡的笔记本计算机通过台式的计算机做代理服务连入网

络中。

- PPP（Point-to-Point）设备支持，其实也就是支持使用电话拨号建立连接时候的通用设备接口。该设备接口使用的点对点（PPP）协议比 SLIP 协议更新、更具有优点。
- SLIP 接口设备，串口的通讯接口设备，在使用串口调制解调器拨号连接时的接口设备。
- Radio network interfaces（无线电网络接口）设备，支持使用业余无线电波通讯协议的网络接口规范设备。
- Ethernet（10 or 100 Mbit）（以太网）设备，也就是平时所说的以太网卡设备，具体支持哪些网卡请参考硬件兼容性列表的说明。
- Token Ring driver support（令牌环）设备，令牌环网络所使用的网卡和以太网的网卡是不同的。也就是需要专门的令牌环网卡才可以，现在使用令牌环的网络主要是一些 IBM 建立的网络。
- FDDI driver support（光纤网络）设备，FDDI 是（Fiber Distributed Data Interface）光纤分布式数据接口的意思，同样这也是指 FDDI 专门的网卡，具体也可以参考有关的说明。

在这些设备中，最主要的就是以太网卡设备了，其次就是 PPP 和 SLIP 设备，当然也不要忘了有用的 PLP 设备。如何配置它们在后面会详细的说明。

1.2 配置网络界面

显然，如果你重新编译了内核，那么你一定有内核的源程序。内核总是要根据源程序中的定义来查找的，所以直接改动内核源程序中使用的网卡的驱动程序中的相应参数为网卡的参数，然后再次重新编译就相当于直接让内核去卡的“位置”寻找。具体如何改动，以及应该改动哪个文件就要靠你自己去研究了，在这里需要提醒的是在改动之前一定要做一个备份。

上面介绍了直接在内核的代码中加入支持网卡的部分，这个办法虽然可行，但是有时候你不一定能找到内核的源代码，或者在编译中有问题而不能成功地完成。有时你不需要让内核在任何时候都支持网卡，那么你可以使用模组来驱动网卡。Linux 系统内核的一个相当好的特点也就是支持了模组的加入。模组在这里就像 DOS 下面的设备驱动程序一样来管理设备，而且模组可以随时地加入到内核中也可以随时地从内核中移出，这是相当方便的，不像在 DOS 或者是 Windows 下，如果要加入一个新的驱动程序，则必须重新引导整个系统。

如果在内核找到了一块网卡，而且有第二块以上的网卡需要支持，使用模组来完成是一个比较理想的选择。当然你也可以用模组或者不使用模组来支持任何一块网卡设备，前提条件是这些网卡占用计算机的资源不产生冲突。

在使用模组来支持网卡之前，你必须知道网卡需要哪个模组来驱动，比如 NE2000 及其兼容的网卡使用 ne.o 模组来驱动；也可以先看看启动模组配置文件/etc/rc.modules 中网卡模组块中的模组都有哪些，或许你一看其名称就知道应该使用哪个模组了。确定

了使用那个模组后，你只需要在/etc/rc.d/rc.modules 中把相应的模组前面的注释符号给去掉就完成了工作。注意，如果是使用NE2000 及其兼容的网卡，那么你一般来说是需要指定 IO 的。

1.2.1 动手配置网络界面

内核找到了网卡设备以后，只不过是系统具有了管理该设备的能力，连入网络还需要设置网络的相关信息。在着手配置之前，必须清楚地知道计算机所使用的网络配置参数，这些参数主要有：

- 使用的 IP 地址，比如 166.111.69.40；
- 网络屏蔽字，比如 255.255.254.0；
- 主机的名称，比如 info；
- 所在网络的域名，比如 cs.tsinghua.edu.cn；
- 主机的默认网关，比如 166.111.68.1；
- 使用的域名服务器地址，比如 166.111.68.2。

在知道了上面的数据以后，可以运行一些现成的网络配置 script 程序来进行配置。在 Slackware Linux 下就可以使用 netconfig 来配置，而在 Red Hat Linux 下面由 netcfg 可以配置（同时编辑 Red Hat 的/etc/sysconfig/下面的 network 文件和 network-scripts 目录下面的配置文件也可以改变网络的配置参数）。使用这些现成的配置办法不需要其他的知识，应该可以顺利地完成配置工作。但是如果有两块以上的网卡，或需要在配置的时候获得更多的灵活性，或希望可以随时地改变一些参数，比如需要时常改变 IP 地址或者加入一条路由，那么还是有必要知道到底应该使用什么方法来配置网络。

1.2.2 ifconfig 配置网络界面

ifconfig 命令也就是 interface config 的缩写，是 Linux 系统中比较通用的一个用来配置网络界面的工具程序。其主要的作用就是将内核已经找到的网络设备（就是在/proc/net/dev 文件中有的）和该设备对应的一个网络地址对应起来，同时也给出了该设备的其他一些特征参数，比如网络屏蔽字等。简单的说，就是把网卡硬件设备和网络地址等参数绑定起来，其实使用 netconfig 一类的程序配置网络的时候，也使用了 ifconfig 命令。

输入 ifconfig 可以列出当前网络界面的配置情况。如图 1.3 所示。

图 1.3 中屏幕上显示出共计 4 个网络设备及其配置情况。

（1）lo 为 Loopback 网络环路设备，任何送向该网络界面的数据报马上返回本地主机，也就是自己向自己发送请求。该界面的作用主要是使一些网络应用程序在网络没有实际建立的时候进行测试。例如在没有实际网络连接的时候可以使用 telnet 127.0.0.1 来连接自己的主机，还可以从上面的显示看出来该设备使用的 IP 地址为 127.0.0.1，网络广播字为 127.255.255.255，网络屏蔽字为 255.0.0.0。下面的是一些该界面接收和发送数据报的统计数字。

（2）eth0 为以太网设备，其硬件地址为 08：00：09：61：32：44，使用的 IP 地址为 10.23.1.1，网络广播字为 10.23.1.255，网络屏蔽字为 255.255.255.0。然后是数据报收发情况，最后是该网络设备（也就是网卡）使用的系统资源情况，使用 IRQ5，IO0x300，

```
[root@cc /root]# ifconfig
lo    Link encap: Local Loopback
      inet addr:127.0.0.1  Bcast:127.255.255.255  Mask: 255.0.0.0
      UP BROADCAST LOOPBACK RUNNING MTU: 3584 Metric: 1
      RX packets:0 errors : 0 dropped: 0 overruns: 0
      TX packets:0 errors: 0 dropped: 0 overruns: 0
eth0  Link encap: Ethernet  HWaddr 08:00:09:61:32:44
      inet addr:10.23.1.1  Bcast: 10.23.0.255  Mask: 255.255.255.0
      UP BROADCAST RUNNING MULTICAST MTU:1500  Metric :
      RX packets:0 errors: 0 dropped: 0 overruns: 0
      TX packets:0 errors: 0 dropped: 0 overruns: 0
      Interrupt : 5 Base address : 0x300  DMA chan: 5
eth1  Link encap: Ethernet  HWaddr 00:00:B4:81:34:33
      inet addr:10.23.0.152  Bcast: 10.23.0.255  Mask: 255.255.255.0
      UP BROADCAST RUNNING MULTICAST MTU:1500  Metric : 1
      RX packets:514 errors: 0 dropped: 0 overruns: 0
      TX packets:300 errors: 0 dropped: 0 overruns: 0
      Interrupt : 10 Base address : 0x240
plip1 Link encap: Ethernet  HWaddr FC:FC:0A:17:02:02
      inet addr:10.23.2.2  P-t-P: 10.23.2.1  Mask: 255.255.255.0
      UP BROADCAST RUNNING NOARP MTU:1500  Metric : 1
      RX packets:0 errors: 0 dropped: 0 overruns: 0
      TX packets:0 errors: 0 dropped: 0 overruns: 0
      Interrupt : 7 Base address : 0x378
[root@cc /root]#
```

图 1.3 ifconfig 命令显示当前网络界面配置

DMA5。

(3) ethl 为第 2 块网卡对应的以太网设备，其相应情况可参考 eth0 的说明。

(4) plip1 为用并口模拟出来的以太网设备，注意它也有硬件地址，一般来说网卡的硬件地址是固化在网卡的 ROM 中的，也就是网卡自带的，但是并口没有自带的硬件地址，也就是说该硬件地址是模拟的时候加给它的，其值为 FC：FC：0A：17：02：02。后面的东西大致相同，只是 inet addr：10.23.2.2 表示它现在使用的网络 IP 地址为 10.23.2.2。而后面的一条：P-t-P：10.23.2.2 表示它现在和另外一个 IP 地址为 10.23.2.1 的网络设备建立了 P-t-P 连接。当然，10.23.2.2 的网络设备就是一条并行线上连接的另外一台计算机的并口模拟出来的设备。最后可以从 Interrupt：7 Base address：0x378 上看出来该设备确实是并行端口其使用的 IRQ 和 IO 都是标准并行端口使用的。

配置设备可以使用：

ifconfig Device _ Name aa.bb.cc.dd netmask 255.255.255.0

其中，Device _ Name 为设备名称，也就是/proc/net/dev 中的设备名。

aa.bb.cc.dd 为该设备的 IP 地址，比如现在配置为 166.111.69.40。

255.255.255.0 为网络屏蔽字，比如现在使用的为 255.255.254.0。

用以上参数配置 eth0 设备，可以使用：

ifconfig eth0 166.111.69.40 netmask 255.255.254.0

结果如图 1.4 所示。

最后使用了 ifconfig eth0 命令来确认 eth0 的配置改变。

如果是建立点对点的连接，例如使用 plipl 设备，那么可以使用：

```
[root@cc /root]# ifconfig eth0 166.111.69.40 netmask 255.255.254.0
[root@cc /root]# ifconfig eth0
eth0      Link encap: Ethernet  HWaddr 08:00:09:61:32:44
          inet addr:166.111.69.40  Bcast: 166.111.68.255
                    Mask: 255.255.254.0
          UP BROADCAST RUNNING MULTICAST MTU:1500
                    Metric : 1
          RX packets:0 errors: 0 dropped: 0 overruns: 0
          TX  packets:0 errors: 0 dropped: 0 overruns: 0
          Interrupt : 5 Base address : 0x300  DMA chan: 5
[root@cc /root]#
```

图 1.4　用 ifconfig 配置 eth0 设备

ifconfig plipl 10. 23. 2. 2 pointtopoint 10. 23. 2. 1 netmasd 255. 255. 255. 0

这样就为 plipl 设备配置了 10. 23. 2. 2 的网络 IP 地址，并且建立的是和 10. 23. 2. 1 的点对点连接。

另外，如果需要暂时禁用一个网络设备的网络界面，例如让 eth0 设备“失效”，那么可以使用 ifconfig eth0 down 来实现；同时在暂时禁用后，还可以通过 ifconfig eth0 up 来恢复原来的使用情况。

使用 ifconfig 配置完网络设备的参数以后，只不过是告诉了内核该设备使用这些参数，如果想要真正地加入网络，那么还需要告诉内核一条到该网络的路由才能够让内核处理网卡需要收发的数据报，而告诉内核路由需要使用的命令是 route 命令。

1. 2. 3　route 命令加入路由

内核需要路由的信息才可以正确地处理网络请求，route 命令是 Linux 系统中比较通用的加入路由的工具程序。route 命令的使用方法比较复杂，但总结起来讲，完成一些普通的配置也比较简单。例如单独地使用 route 命令显示当前的路由设置情况，如图 1. 5 所示。

```
[root@cc /root]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.23.0.0       *               255.255.255.0   U     0      0        3 eht1
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
default         cc              0.0.0.0         UG    0      0        0 eht1
[root@cc /root]#
```

图 1.5　显示当前路由表

如果要向内核路由表中增加一条路由，使用的命令格式为：

route add -net Net _ address netmask 255. 255. 255. 0 [Device _ name]

route add -host Host _ address [Device _ Name]

其中，route 后面的第一个参数 add 表示向内核路由表增加一条路由，如果这个参数替换为 del，那么表示将某条路由删除。

第二个参数如果是-net 表示是增加一条网络路由；如果是-host 表示仅仅是一条指向

另外一台主机的路由。一般情况下，如果是在网络中都增加一条网络路由，而指向一台主机的路由一般用在点对点的连接方式中。

如果是使用-net 加入网络路由，那么后面跟的 Net _ address 也就是网络地址，网络地址的数值由本地主机 IP 和网络屏蔽字求与来得到。如果使用-host 加入指向主机的路由，那么 Host-address 就是对方主机的 IP 地址。

对于-host 加入主机路由，是不需要网络屏蔽字的，因为屏蔽字的主要目的是表征该主机是处于哪个网络段中的，所以加入指向一台主机的路由根本不需要屏蔽字。如果用-net 来加入网络路由，那么输入屏蔽字是一个好的习惯，虽然在没有输入屏蔽字的情况下 route 会根据加入路由的设备所使用的屏蔽字来决定屏蔽字的数值，即使用在 ifconfig 中已经指定的屏蔽字。

最后的参数是需要加入路由的设备名，也就是像 eth0，eth1，plip1 等。在系统除 lo 外只有 eth0 设备时，可以省略这个参数，route 会自动地把路由加到 eth0 这个设备上；但是，如果不是 eth0 设备，或需要在其他的设备上加入路由，那么就不能省略设备名，否则会报告错误而无法正确地加入路由。

加入了到网络的路由后应该可以连接到子网络中的其他主机。使用 ping 命令可以查看是否已经正确连通。如果没有正确连通，应该再检查一下上述的步骤是否有误。如果已经正确连通，则需要在内核路由表中加入默认网关的路由信息，以通过网关连接到子网以外的网络，即对于不在同一个网络地址中的主机的网络请求都送向网关，由网关处理这些请求，并转发到外面。加入默认网关的命令是：

route add default gw Gateway _ IP [Device _ Name]

其中，Gateway _ IP 的值为默认网关的 IP 地址，如果只有 eht0 设备，那么可以省略 Devce _ Name 参数，否则还是需要指定一下。当然，如果不特别指定是哪个网络设备使用该默认网关，那么 route 也可能在不冲突的情况下通过网络地址和网关的地址来确定是哪一个设备。例如，给 eth0 这个设备配置一个默认的网关为 166.111.68.1 使用的命令为：

route add default gw 166.111.68.1 eth0

到此为止，计算机的网络算是基本配置完成了。可以尝试使用 ping 来查看是否能连接到外部的主机了。域名服务的配置，可以更改/etc/resolv.conf 文件的内容为 DNS 服务。

1.3　有关网络的常用命令

在了解和使用网络之前必须知道一些常用的网络命令，这样才可以在实际的网络应用中进行更灵活的控制。

1.3.1　arp

用 arp 命令可以显示系统的地址解析协议的缓冲内容。其参数如表 1.1 所示。

表 1.1 arp 命令的参数

参　数	描　述
-v	详细说明方式
-n	显示 IP 地址
-H type	
-a hostname	显示指定主机名的网络设备情况，如果未指定，显示所有设备
-d hostname	删除指定主机的 Arp entry
-D	使用网卡的硬件地址
-i if	
-s hostname hw _ addr	手动增加 arp 的 IP mac 对应地址
-f filename	从文件读入 arp 设置

1.3.2 hostname

用 hostname 命令可以显示或者设置系统的主机名。其参数如表 1.2 所示。

表 1.2 hostname 命令的参数

参　数	描　述
-a	显示系统的别名
-d	显示 DNS 域名
-f	从指定文件读取主机名
-h	显示 hostname 的帮助信息
-i	显示主机的 IP 地址
-s	显示主机名的简单名字，如 info. tsinghua. edu. cn 就显示 info
-V	显示 hostname 的版本信息
-v	详细报告并说明过程
-y [NIS/YP name]	显示或者设定 NIS/YP 域名

1.3.3 netstat

netstat 命令用于显示网络连接、路由表、网卡统计和匿名连接。其参数如表 1.3 所示。

语法：

```
netstat [-enc] {——maxquerade|-M}
netstat {——statinstics|-s}
netstat {-V|——version} {-h|——help}
```

表 1.3 netstat 命令的参数

参 数	描 述
无选项	从监听到的包来查询网络连接情况，如果没有设定地址的范围，所有连接都将被显示。使用-e 参数可以得到用户帐号的信息，-v 参数可以得到不支持的地址族的列表，-o 显示网络时钟的信息，-a 显示所有的 socket 的信息。
-r	可以获得内核路由表的信息
-i	获得所有或者指定网卡的信息

1.3.4 login

login 命令用于登录一个主机，也可以用来从一个用户转到另外一个用户，这一点有点儿像 su 命令，但是它的用法比 su 要多一些。其参数如表 1.4 所示。

语法：

login [参数] [主机名] [用户名]

表 1.4 login 命令的参数

参 数	描 述
用户名	如果没有任何参数，则 login 需要一个用户名
-p	保留环境变量
-f	忽略第二次登录证明，这对 root 无效
-h	超级用户远程登录主机

1.3.5 nslookup

nslookup 命令交互式地查询 Internet 域名 IP 对应服务。这个命令有以下两种模式。

1. 交互式

允许用户查询不同的主机和域名或者显示一些主机的域名信息；它有两种情况：

如果没有给出参数，缺省的域名服务器将被使用。

如果第一个参数是“－”而且第二个参数是主机名或者是 Internet 的域名服务器的地址。

2. 非交互式

当要搜寻的 Internet 主机的名字作为第一个参数被给出则使用非交互模式，第二个参数可以用来指定域名服务器，非交互模式只输出主机的名字和一些要求的信息。

1.3.6 ping

ping 命令用于检查两台联网的计算机之间的物理网络是否连通，这个命令通常用于检查网络线路是否存在物理故障。其参数如表 1.5 所示。

ping 命令向网络主机中发送 ICMP 包，并要求目标主机响应。

表 1.5 ping 命令的参数

参　数	描　述
-c	收到主机响应即停止
-f	洪水式发包，一旦接收到主机响应就发包或每秒数百个包，通常用于快速的检查丢包率，只有超级用户才可以使用本参数
-i	每发一个包等待一秒钟
-l	强制以最快速度发包，只有超级用户才能用
-n	只显示远程主机的 IP，不显示域名
-p pattern	可以指定所发 16 位包的内容
-q	除了起始和结束报告之外，不显示其他报告
-s packetsize	自定义包的大小

1.3.7 route

route 命令显示或者设定 IP 路由表。route 命令设定内核的路由表，这个命令的主要用途是用来在使用 ifconfig 命令后，设定网络或者路由地址设定的问题。其参数如表 1.6 所示。

语法：

route [参数] [主机名] [gw 网关名]

表 1.6 route 命令的参数

参　数	描　述
-v	详细显示
-n	只显示主机的 IP 地址而不显示域名
-e	显示路由表
-net	目标是一个网址里面查找
-host	是一个主机地址
不加任何参数	显示内核的路由表
add	加入一个路由
target	目标网络或者是主机，既可以输入 IP 地址也可以输入域名
netmask Nm	设定并加入路由的掩码，这个选项只对网络路由有效
gw Gw	每一个发往目标网络或者主机的 IP 包都要通过网关路由器

1.4 小　结

本章主要介绍了如何在 Linux 系统下配置网络以及与网络相关的一些常用命令。

第 2 章　TCP/IP 的基本原理

数据通信目前已经成为计算机技术的一个基本组成部分，而且随着世界经济一体化的进程，网络互连将会成为人们生活中一个不可缺少的重要组成部分。世界范围内的网络把诸如气象条件、农作物生产和飞机航班等如此多样化的数据集合在一起；工作组建立电子邮件表，以便于分享大家共同感兴趣的信息；业余爱好者为它们的家庭计算机交换程序；因为数据网络允许科学家把数据和程序发送到远程超级计算机上进行处理、检索结果，以及使他们能与同事交换信息，所以网络已经成为人们生活中不可缺少的一部分了。

2.1　计算机网络的体系结构模型

提到计算机网络，就不得不先介绍一下其结构模型。正如计算机程序一样，为了使编写、调试、编译程序更容易、清晰、简洁，并且易于管理，我们通常将程序按功能在结构上分成各个模块；为了使计算机通信具有硬件透明性（即独立于某种具体硬件）、易于管理网络和解决硬件失效、网络拥塞、数据损伤、数据重复及乱序等问题，我们将计算机网络协议在概念上分层，不同的协议层担任不同的任务。目前在计算机网络中主要有两种体系结构模型：一是 ISO 模型，二是 TCP/IP 模型。

2.1.1　ISO 模型

ISO 模型是由国际标准化组织（International Standardization Organization）制作的，也就是开放式系统互连的参考模型。如图 2.1 所示。

虽然 ISO 模型的层次结构的设计初衷是提供概念性模型而不是作为具体的实现指导，但是它也成为了若干协议实现时的参考。在这些以 ISO 模型作为实现参考的协议中，最著名同时也是应用最广泛的要算是 X.25 协议了。研究 X.25 协议有助于理解 ISO 的分层思想。

在 X.25 协议看来，网络的运行很像一个电话系统。一个 X.25 网络被设计为由复杂的、具有选择分组路由能力的分组交换机来构成。主机并非直接连到网络的通信线路上，而是使用一条串行通信线与一个分组交换机相连。从某种意义上说，主机与 X.25 分组交换机的连接构成了一个小型的、由一条串行线组成的网络。主机必须遵从复杂的操作流程才能将分组传送到网络上。

1. *物理层*

X.25 定义了主机与分组交换机之间物理连接的标准，也定义了两台机器之间分组传输的流程。在 ISO 参考模型中，第 1 层定义了物理连接的标准，其中包括对电压和电流等电气特性值的规定。相应的 X.21 协议给出了公用数据网上所遵从的细节规定。

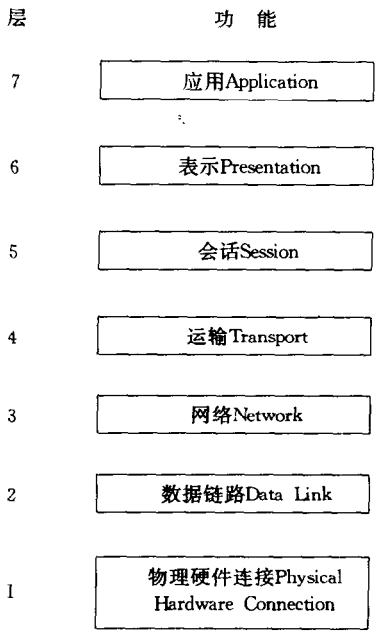


图 2.1 协议软件的ISO七层参考模型

2. 数据链路层

X.25 标准的第二层定义了如何在主机和分组交换机之间传输数据。X.25 使用了术语“帧”来描述在这两者之间传输的数据单元（请注意，X.25 定义的帧与我们曾用过的帧的意义有细微的差别）。由于底层的硬件只能传送比特流，第二层协议必须定义帧的格式，给出帧边界的识别方法。由于传送时的差错会破坏数据，第二层协议还包括了差错检测机制（如帧的校验和）。最后，由于传输是不可靠的，第二层协议定义了互相交换确认的机制，使得两台机器能够知道帧是否传输成功。

3. 网络层

ISO 参考模型指出，第三层协议要对主机和网络之间的交互进行定义，第三层又叫通信子网层，它定义了在网络上传输的基本数据单元以及目的寻址和选路的概念。在 X.25 协议中，主机和分组交换机之间的通信与实际的通信在概念上是独立的，网络上遵从第三层协议的分组可以比在第二层上传输的帧大一些。第三层软件按照网络所期望的形式装配其分组，再使用第二层把它传送给分组交换机（多半要把分组分为小块）。第三层还要处理网络拥塞的问题。

4. 传输层

第四层提供收信主机到发信主机之间的端到端的可靠传输。需要指出的是，即使下层的协议在传输时都提供可靠性检查，作为端到端的第四层也要进行检验来确保在中间的传输过程中没有机器出故障。

5. *会话层*

ISO 模型中的高层描述了协议软件应该如何组织，以便提供应用程序所需的功能。

6. *表示层*

ISO 的第六层的目的在于将许多应用程序在使用网络时所需的功能包容进来。典型的例子有将文本压缩，及将图像转换为可在网络上传输的比特流。

7. *应用层*

ISO 的第七层包括了使用网络的应用程序，例如电子邮件和文件传输程序等。

2.1.2　TCP/IP Internet 分层模型

第二个重要的分层模型并非出自哪个标准组织，而是出自一些对 TCP/IP 协议的研究稍作改动而形成的，ISO 的参考模型就可以用于描述 TCP/IP 的层次结构，但是这两者之间存在本质性的区别。

TCP/IP 模型是由四个构筑在第五层即硬件层上的概念性层次构成的。图 2.2 给出了这些概念性的层次结构以及在这些层次之间传输的数据形式。

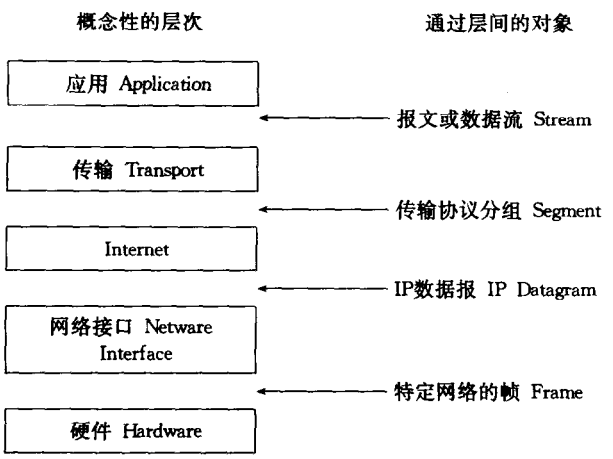


图 2.2　TCP／IP的四个概念性层次几个层之间传输的对象形式

1. *应用层*

在这个层中，用户调用应用程序来访问 TCP/IP 互连网络，与各个传输层协议协调工作的应用程序负责接收和发送数据。每个应用程序选择适当的运输服务类型（服务包括独立的报文序列和连续字节流两种类型）。应用程序把数据按照传输层的格式要求组织好后向下层传送。