Linux 网络结构及实现分析

黄 杰 （南京邮电学院 通信工程系 210003）

摘要：本文比较详细地分析了Linux网络结构及其实现，并相应指出了其在Linux源代码中的文件位置。

关键词：Linux 内核 TCP/IP 操作系统

Linux是开放源码的操作系统，它具有强大的网络功能。Linux的网络实现是以4.3 BSD为模型的，它支持BSD Sockets(及一些扩展)和所有的TCP/IP网络。选这个编程接口是因为它很流行，并且有助于应用程序从Linux平台移植到其他Unix平台。

一、Linux TCP/IP 网络层

如网络协议本身的层次，Linux的网络也是分层实现的。如图1所示。

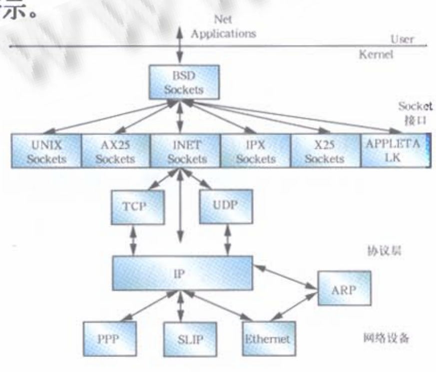

图 1 Linux 网络层结构

从图1可以看出Linux用一系列相互连接的软件层实现Internet协议地址族。BSD套接字(BSD Sockets)由专门处理BSD Sockets通用套接字管理软件处理。它由INET Sockets层来支持，这一层为基于IP的协议TCP和UDP管理传输端点。UDP和TCP模块实现UDP、TCP协议。IP层实现Internet协议的代码。这些代码给要传输的数据加上IP头，并知道如何把传入的IP包送给TCP或UDP。在IP层以下，是网络设备来支持所有Linux网络工作，如PPP和以太网。网络设备不总是物理设备，一些象loopback这样的设备是纯软件设备。

ARP协议位于IP层与支持ARP的协议之间。

二、BSD Sockets 接口

1. 概述

这是一个通用的接口，它不仅支持各种网络工作形式，而且还是一个交互式通信机制。一个套接字描述一个通信连接的一端，两个通信程序中各自有一个套接字来描述它们自己那一端。套接字可以被看成一个专门的管道，但又不象管道，套接字对它们能容纳的数据量没有限制。Linux支持多种类型的套接字，每一类型的套接字有它自己的通信寻址方法。Linux支持下列套接字地址族或域：UNIX、INET、AX25、IPX、APPLLETALK、X25。这当中，以INET地址族最为常用。本文主要介绍INET地址族。

另外，Linux BSD套接字支持下列多种套接字类型：Stream、Datagram、Raw、Packet等。

2. 具体实现

实现BSD Sockets的主要文件有:linux/net/protols.c、socket.c、sysctl_net.c。其数据结构如图2所示。

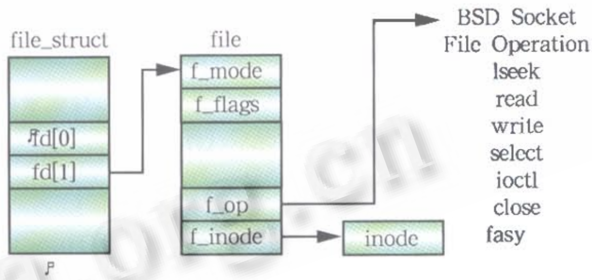

图 2 Linux BSD Sockets 数据结构

需要注意的是对BSD sockets进行准确操作要依赖于它下面的地址族。与虚拟文件系统一样，Linux从BSD sockets层抽象出socket接口，应用程序和BSD sockets由每个地址族的特定软件来支持。内核初始化时，地址族被置入内核中并将自己注册到BSD sockets接口。之后，当应用程序建立用使用BSD sockets时，在BSD sockets与它支持的地址族之间将产生一个联接。这一联接是由交叉链接数据结构和地址族表特定支持程序产生。

构造内核时，一些地址族和协议被置入protocols向量。每个由它的名称来表征，例如，“INET”和它的初始程序地址。当套接口启动时被初始化时，要调用每一协议和初始程序。对socket地址族来说，这会导致它们注册一套协议操作。这是一套例程，其中的每一例程执行一个

© 中国科学院软件研究所 http://www.c-s-a.org.cn

特定的针对那一地址族的操作。已注册的协议操作被存在pops向量(一个指向proto_ops数据结构的向量)中。

proto_ops结构由地址族类型和一系列指向与特定地址族对应的socket操作例程的指针组成。pops向量通过地址族标识符来索引，如Internet地址族标识符(AF_INET是2)。

三、INET Sockets层

根据协议分层的思想，一个协议使用另一个协议的服务。INET sockets层支持包括TCP/IP协议在内的Internet地址族，它与BSD socket层的接口要通过一系列Internet地址族socket操作。这一操作是在网络初始化时就已经注册到BSD sockets层的(保存在pops向量中)。BSD sockets层从已注册的INET proto_ops数据结构中调用INET层sockets支持例程来为它执行工作。例如，一个地址族为INET的BSD sockets建立请求，将用到下层的INET sockets的建立函数。在这些操作中，BSD sockets层把用来描述BSD sockets的socket结构传构到INET层。为了不把BSD sockets与TCP/IP的特定信息搞混，INET sockets层使用它自己的数据结构sock，它与BSD sockets结构相连。它用BSD sockets的data指针来连接sock结构与BSD sockets结构。这意味着后来的INET sockets调用能够很容易地重新找到sock结构。sock结构的协议操作指针也在初始化时建立，它依赖与被请求的协议。如果请求的是TCP，那么sock结构的协议操作指针将指向TCP连接所必需的TCP协议操作集。

也就是说当系统调用建立socket、bind socket、connect、listen、accept等BSD Sockets操作时，由BSD Sockets层传给INET Sockets层处理。

与INET Sockets实现对应的文件主要是linux/net/ipv4/af_inet.c。

四、IP层

1. Socket缓存

在分层网络结构中，每一层协议使用另外层提供的服务，所以使用多层网络协议会有一个问题：每个协议都要在传送数据时加上协议头和协议尾，而数据到达时又要将之去掉。这样，在不同的协议间要有数据缓存，每一层需要知道特定协议的头和尾放在哪个位置。一个解决办法就是在每一层中都拷贝缓存，但这样做效率就很低。Linux用Socket缓存或者说sk_buff来在协议层与网络设备驱动之间交换数据。sk_buff是linux网络数据处理的核心数据结构。系统提供了对sk_buff数据结构的标准操作函数。因此每个协议层就可以通过这些标准的函数来操作应用程序数据。

与sk_buff实现对应的文件主要是：linux/net/core/sk_buff.c。

2. 接收IP包

Linux中，用device数据结构表示设备，每个device结构描述了它的设备并提供回调例程，当需要网络驱动来执行工作时，网络协议调用这些例程。当一个网络设备从网上接收数据包时，它把接收的数据转换成sk_buff结构。这些sk_buff被网络驱动程序加入到了backlog队列中等待处理。而这一系列device数据结构有在dev_base链表中相互连接起来。网络底层通过ptype_all表或ptype_base表来决定由那个协议处理接收传入的网络数据包。

3. 发送IP包

应用程序交换数据时要传输包，否则由网络协议在建立连接或支持一个已建立的连接时来生成。无论数据是由哪种方法生成的，都要建立一个sk_bufF来包含数据，当通过协议层时，这些协议层会加上各种头。sk_buff需要通过网络设备传输。首先协议(如IP)需要确定是哪个网络设备在用。这有赖于包的最佳路由。对于通过modem连入一个简单网络(如通过PPP协议)的计算机来说，路由的选择是很简单的；此时数据包应该通过本地环路设备发送给本地主机，或发送给PPP modem连接的网关。对于连在以太网上的计算机来说，连接在网络上的计算机越多，路由越复杂。

对于每一个被传输的IP包，IP用路由表来为目的IP地址解析路由。从路由表中成功地找到目的IP时将返回一个描述了要使用的路由的rtable结构。这包括要用到的源IP地址，网络device结构的地址，有时还有预建立的硬件头。这些硬件头是网络设备特定的，包含了源和目的的物理地址和其他的特定媒质信息。如果网络设备是一个以太网设备，源和目的地址应是物理的以太网地址。硬件头在路由的时候会缓存起来，因为必须将它加到每一个要传输的IP包中。硬件头包含的物理地址要用ARP协议来解析。传出的包在地址被解析后才会发出。解析了地址后，硬件头被缓存起来以便以后的IP包在使用这一接口时不需要再使用ARP。

需要注意的是，在发送接收IP包时可能需要IP包的

© 中国科学院软件研究所 http://www.c-s-a.org.cn

分割与组装。与IP包收发对应的文件主要是: linux/net/ipv4/ip_input.c、ip_output.c。

五、地址解析协议（ARP）

地址解析协议担当了一个把IP地址翻译成物理硬件地址(如以太网地址)的角色。IP在将数据（以sk_buff的形式)通过设备驱动传送时需要这一转换。它执行各种检查，来看是否这一设备需要硬件头，是否需要重建包的硬件头。Linux缓存了硬件头，这样可以避频繁重建。如果需要重建硬件头，则调用设备指定的硬件头重建例程。所有的以太网设备使用相同的头重例程，这些例程将目的IP地址转换成物理地址。

ARP协议本身是很简单的，它包括两个消息类型: ARP请求与ARP应答。ARP协议在Linux中是围绕arp_table结构表来建立的，每个结构描述一个IP到物理地址的转换。这些表项在需要进行IP地址解析时生成，在随时间变旧时被删除。

与ARP实现有关的文件是linux/net/ipv4/arp.c。

六、IP路由

路由是一个非常复杂的问题，在此就不详细介绍。只是我们应知道的是Linux内核支持路由协议，它实现了RIP路由协议。

七、结束语

Linux的一个非常诱人的优点是开放源代码，加之我们正处于网络时代，所以对Linux操作系统的网络功能实现的研究很有必要。无论是教学实践还是科研开发都需要这些方面的知识。■

参考文献

1 *Linux HOWTO文档.*

2 *李善平 郑扣根. Linux操作系统及其实验教程. 机械工业出版社. 1999.*

© 中国科学院软件研究所 http://www.c-s-a.org.cn