说 明 书 摘 要

基于 libpcap 的计算机网络原理教学系统，属于计算机应用技术，具体涉及实现计算机网络原理中典型协议形成的协议栈和 Socket 接口层模块的方法。该方法能够帮助学生提高动手能力，理解典型的计算机网络协议，掌握计算机网络知识，进而提升计算机网络原理课程的教学质量。本发明所述系统包括驱动层模块，数据链路层模块，邻居子系统模块，网络层模块，路由子系统模块，传输层模块，Socket 接口层模块。本发明结合抓包分析，要求学生自行编程实现典型计算机网络协议，以这种方式加深学生对计算机网络原理的理解和认识，同时提高学生的动手实践能力。本发明利用面向对象编程模型，采取典型协议分离的方式，降低了整体实现难度，让学生能够一步一步地实现单一协议，逐渐地掌握整个系统，从而掌握计算机网络原理。

权 利 要 求 书

1. 本发明所述的基于 libpcap 的计算机网络原理教学系统，包括驱动层模块，数据链路层模块，邻居子系统模块，网络层模块，路由子系统模块，传输层模块以及 Socket 接口层模块；

（A）驱动层模块，基于 libpcap 直接从网卡收发数据包，不经过操作系统网络子系统；对于接收，将从 libpcap 收到的数据包交给数据链路层模块进一步处理，对于发送，利用 libpcap 发送接口，将数据链路层待发送数据包转化为字节缓冲区作为参数，进行发送；

（B）数据链路层模块，根据接收到的数据包的帧类型，去掉数据链路层头部以后分发给网络层（IP）模块，或邻居子系统（ARP）模块；对于待发送数据包，加上数据链路层头部信息并调用驱动层接口发送；

（C）邻居子系统模块，向网络层提供发送接口，维护一个网络地址到硬件地址映射关系的缓存表，和一个缓存数据包的待发送队列；发送数据包时，对于缓存表中已有的映射，直接调用数据链路层接口发送该数据包；对于未找到映射的数据包，放入待发送队列，并发送一个 ARP 请求，当接收到回复时添加该映射到缓存表，同时处理与该映射对应的待发送队列；收到 ARP 请求的主机也对该映射进行缓存；

（D）网络层模块，根据接收到的数据包目的网络地址是否为本机地址分别处理，若是，则进一步解析网络层头部，当数据包不是分片时，去掉网络头部后直接交由传输层协议 UDP 或 TCP 进一步处理，当数据包是分片时，添加该分片到重组链表，如果此分片为最后一个分片，即可完成重组；若为非本机数据包，同时配置了允许路由转发，则调用路由子系统进一步处理；在发送数据包时，如果数据包大小超过 MTU，则需要进行分片，每个分片都有自己的头部；添加网络层头部信息后，调用邻居子系统模块发送数据包；

（E）路由子系统模块，运行路由算法，保存路由表，根据数据包目的地址选择路由，决定下一跳地址；

（F）传输层模块，收到数据包后，首先对数据进行完整性校验，这是通过校验和实现的；发送数据包时，先添加传输层头部信息，再调用网络层接口进一步处理。传输层根据 RFC 标准实现用户数据报协议 UDP 和传输控制协议 TCP；

（G）Socket 接口层模块包括协议栈 Socket 单元和应用程序 Socket 单元，Socket 接口层模块作为协议栈和应用程序的桥梁，通过共享内存和信号的方式进行数据交互；协议栈 Socket 单元和应用程序 Socket 单元各自拥有一块内存缓冲区，写数据时写入到自己的缓冲区，读数据时从对方的缓冲区读取；发送数据时先封装到约定的数据结构，再写入缓冲区，然后通过信号通知对方接收；接收数据时按照约定的数据结构对待读缓冲区数据进行解析，然后作进一步处理，完成接收。

2. 所述的基于 libpcap 的计算机网络原理教学系统，其特征在于：使用虚拟机作为系统运行平台，减少环境配置的麻烦，避免外界网络数据的干扰；采用典型协议分离的实现方式，支持将单一协议正确实现后加入系统，以恢复系统完整。

基于 libpcap 的计算机网络原理教学系统和方法

技术领域

本发明涉及计算机网络技术和教育领域，尤其涉及的是，一种基于 libpcap 的计算机网络原理教学系统和方法。

背景技术

当代计算机技术发展迅猛，近年来，人们对云计算，大数据和网络安全的关注度居高不下，而这些技术的发展都离不开计算机网络技术的支撑。熟练地掌握计算机网络原理，是计算机科学与技术和与之专业相关的学生必须具备的能力，但计算机网络体系庞大，结构复杂，一直以来学生们都感觉计算机网络原理这门课程难以掌握。各高校对计算机网络课程虽有重视，增加了实验课时，但实际上大部分依然以理论教学为主，或只是做一些简单的网络通信程序，抓包分析实验，学生对知识的理解依然停留在基本的认识上。

培养具有高水平的计算机网络技术人才，是满足企业需求，未来社会信息化，网络空间安全的重要保障。

要掌握好计算机网络原理，不仅需要熟悉关键知识点的每个细节，还要对整体结构有一定的把握。一些好的计算机网络原理教材会在每章最后一节给出实验练习，让学生亲自动手抓包分析，这种方式一般能加深学生宏观上的理解，但不够细致，也不够深入。

为此，需要提供一种计算机网络原理教学系统和方法，既能针对关键知识点的细节，并且层层深入，又能在整体上前后贯穿，提高学生对计算机网络原理的理解和认识，同时提升计算机网络原理课程的教学质量。

发明内容

本发明主要解决的技术问题是针对计算机网络原理体系庞大，结构复杂等实际困难，提供一种针对典型协议和整体结构进行实验教学的方法。

为解决上述技术问题，本发明采用的技术方案是，提供一种基于 libpcap 的计算机网络原理教学系统和方法，所述方法包括以下步骤：构建驱动层，基于 libpcap 直接从网卡接收和发送数据包，封装接口以提供发送数据包功能，将接收到的数据包发往数据链路层；构建所述数据链路层，封装接口以提供发送数据帧功能，将接收到的数据帧发往邻居子系统或网络层；构建所述邻居子系统，封装接口以提供发送网络层报文功能，解析接收到的地址解析包将该映射缓存到地址映射表和处理待发送报文队列，所述待发送报文队列为所述网络层报文；构建所述网络层，封装接口以提供发送报文功能，分析接收到的报文进行路由转发或本机投递，即发往路由子系统或传输层；构建所述路由子系统，根据报文目的地址选择下一跳进行路由，调用邻居子系统发送接口发送报文；构建所述传输层，封装接口以提供发送用户数据功能，将接收到的用户数据发往 Socket 接口层，所述 Socket 接口层为协议栈与应用程序接口层；按

照所述步骤构建形成的协议栈，为完整的计算机网络框架，选择所述任意步骤，或选择任意步骤中某一协议进行实现单独作为实验要求，正确完成该实验即恢复协议栈的完整。

所述系统包括：驱动层模块，作为 libpcap 的包装器，用于接收和发送数据包；数据链路层模块，用于成帧，计算循环冗余校验码，解析帧；邻居子系统模块，用于硬件地址和网络地址之间的转换，维护地址映射表及待发送报文队列；网络层模块，用于报文封装和解析报文，分片，重组；路由子系统模块，用于报文转发；传输层模块，用于网络传输结构的初始化，分用，复用，支持尽力传输服务和可靠传输服务；Socket 接口层模块，用于协议栈和应用程序通信，提供数据接收和发送接口，参数控制接口。

在本发明另一实施例中，该数据链路层采用以太网帧类型。

在本发明另一实施例中，该协议栈和 Socket 接口层运行在不同进程，采用共享内存和信号的方式互相通信。

本发明的有益效果是，通过提供完整的计算机网络协议栈，和分离各典型协议的实现，在要求动手的基础上又降低了实现难度，有助于学生逐步地掌握各个协议，进而在整体上掌握计算机网络原理。

附图说明

为了更清楚地说明本发明实施例中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍。

图 1 为本发明实施例系统整体结构示意图。

图 2 为本发明实施例数据链路层模块结构示意图。

图 3 为本发明实施例邻居子系统模块流程图。

图 4 为本发明实施例网络层模块结构示意图。

图 5 为本发明实施例传输层模块结构示意图。

具体实施方式

为了便于理解本发明，下面结合附图和具体实施例，对本发明进行更详细的说明。附图中给出了本发明的较佳的实施例。但是，本发明可以以许多不同的形式来实现，并不限于本说明书所描述的实施例。相反地，提供这些实施例的目的是使对本发明的公开内容的理解更加透彻全面。

需要说明的是，除非另有定义，本说明书所使用的所有的技术和科学术语与属于本发明的技术领域的技术人员通常理解的含义相同。在本发明的说明书中所使用的术语只是为了描述具体的实施例的目的，不是用于限制本发明。

系统整体结构如图 1 所示，Socket 接口层模块的协议栈 Socket 单元和应用程序 Socket 单元分别运行在不同进程，通过共享内存和信号的方式达到应用程序和协议栈互相通信的目的。共享内存分成两块：buffer1 和 buffer2，其中，buffer1 由协议栈 Socket 单元写入，应用程序 Socket 单元读取，buffer2 由应用程序 Socket 单元写入，协议栈 Socket 单元读取。通过信号保持同步。

第一步，安装虚拟机及操作系统。在宿主机上安装 VirtualBox 4.3 虚拟机，同时下载 Debian

8 操作系统镜像，创建虚拟机 VMA，载入下载的系统镜像进行虚拟机操作系统的安装。再次创建新的虚拟机 VMB，并用下载的操作系统镜像进行安装。此时拥有虚拟机 VMA 和虚拟机 VMB。在两个虚拟机上均安装构建工具 CMake，编译工具 G++，本系统依赖库 libpcap。

第二步，配置虚拟机内部网络。创建 VirtualBox 的内部网络 intnet，启动 VirtualBox 的 DHCP 服务器，并将新创建的两个虚拟机设置为内网模式，加入内部网络 intnet。确保虚拟机之间可以互相 ping 通。可选的，关闭网络子系统 ARP 服务。

第三步，实现驱动层模块。利用 libpcap 封装为所述系统的 Driver 类，包括利用 libpcap 提供的 pcap_findalldevs 接口，检测系统可用的网卡设备，并取得其硬件和网络地址，选择第一个处于活跃状态的以太网卡设备作为默认收发设备；利用 pcap_inject 以提供发送接口；开启新线程，利用 pcap_lookup 以提供接收接口。

第四步，实现数据链路层模块，如图 2 所示，包括发送接口，将从所述网络层模块接收到的数据成帧，然后调用 Driver 发送接口发送数据包；接收接口，从 Driver 接收数据包，若非以太网帧则直接作丢弃处理，否则对数据包进行解析操作，根据该帧的以太网帧类型，将数据包交给相应模块进一步处理，如 ARP 包将分发给邻居子系统模块，IP 包将分发给网络层模块。

第五步，实现邻居子系统模块。当网络层模块进行数据包的发送时，在交给数据链路层模块之前，还需要获取网络地址对应的硬件地址，邻居子系统模块即完成此工作。如图 3 所示，网络层模块调用邻居子系统模块发送数据包，若邻居子系统模块在其缓存表中找到了对应网络地址与硬件地址的映射，则使用该地址直接调用数据链路层模块进行发送；若未找到，则需要缓存该数据包到待发送队列，并发送 ARP 请求。当 ARP 响应到达时，添加该映射到缓存表，并处理对应的待发送队列，将待发送数据包逐一发送。对于接收到 ARP 请求的情况，也对该映射进行缓存。对于下一个同样目的地址的数据包到达时，若对应 ARP 请求还未收到 ARP 响应，不会再次发送 ARP 请求。

第六步，实现网络层模块。对于发送，当网络层模块有待发送数据包时，首先检测数据包大小是否超过网卡设备的 MTU，若没有超过，则添加网络层头部，然后调用邻居子系统模块进行发送；若数据包大小超过 MTU，则需要进行分片处理，同一报文的分片 id 保持一致，分片相对偏移 offset 依据已分片的大小计算，分片信息存放在每个分片的网络层头部相应字段；对于接收，网络层模块从数据链路层模块接收到数据包，根据数据包目的网络地址分别进行处理，如果是发往本机的数据包，则根据网络层头部分片偏移 offset 及 IP_MF 标志判断是否为分片，若两个标志均为零则不是分片，解包后交由传输层模块对应协议进一步处理，否则是分片，缓存该分片到重组队列，当所有分片正确无误到达时，即可恢复完整数据包的重组，解包后交由传输层模块对应协议进一步处理，如果不是发往本机的数据包，同时配置了支持路由转发功能，则调用路由子系统模块进行非本地数据包的转发，若未配置路由转发则丢弃该数据包。

第七步，实现传输层模块。主要实现两种协议，简单数据报协议 UDP 和可靠传输协议 TCP。对于 UDP 协议，发送数据包时添加 UDP 头部，其校验和字段需根据 UDP 伪首部计算，调用网络层模块提供的发送接口进行发送，UDP 不进行分片处理，采用网络层模块的分片处理方式；从网络层模块接收到数据包时，计算校验和以判定数据是否完整一致，然后去掉头部交给协议层 Socket 进一步处理。对于 TCP 协议，发送数据包时添加 TCP 头部，其校验和字段需

根据 TCP 伪首部计算，调用网络层模块提供的发送接口进行发送。与 UDP 协议不同，TCP 协议为了保证数据的可靠传输，在发送数据之前需要进行连接，进行三次握手以确保连接正确建立，数据传输完毕后进行四次挥手以确保连接正常关闭；在数据传输过程中，通过捎带确认和超时重传机制保证数据的可靠性，通过发送窗口和接收窗口达到拥塞控制和流量控制的目的。

第八步，实现 Socket 接口层模块，包括 socket(),bind(),listen(),accept(),sendto(),recvfrom(),send(),recv(),close()在内的标准 socket 接口函数，完成应用进程数据发送和接收。Socket 接口层模块包括协议栈 Socket 单元和应用程序 Socket 单元，分别运行在不同进程，通过共享内存和信号的方式达到应用程序和协议栈互相通信的目的。共享内存分成两块：buffer1 和 buffer 2，其中，buffer1 由协议栈 Socket 单元写入，应用程序 Socket 单元读取，buffer2 由应用程序 Socket 单元写入，协议栈 Socket 单元读取。通过信号保持同步，Socket 创建，监听，关闭等请求使用 SIGUSR1 信号，数据发送和接收使用 SIGUSR2 信号，协议栈 Socket 单元总是向应用程序 Socket 单元回复从应用程序 Socket 单元收到的相同信号。

第九步，分别使用操作系统的网络子系统和本发明实现的网络系统编写网络通信程序 datetimedemo 和 echodemo。其中，datetimedemo 采用传输层 UDP 协议，客户端发送任意内容，如“time”，服务端回复当前时间；echodemo 采用传输层 TCP 协议，客户端发送任意内容，如“hello”，服务端回复相同内容，“hello”。

第十步，以正确实现协议层中典型协议之一作为实验要求，保持协议层中其他模块不变，形成计算机网络原理教学实验之一。对各典型协议均执行此步骤，形成计算机网络原理教学的所有实验。

本发明从实际例子出发，通过提供完整的计算机网络协议层，和分离各典型协议的实现，在要求动手的基础上又降低了实现难度，有助于学生逐步地掌握各个协议，进而在整体上掌握计算机网络原理。

说 明 书 附 图

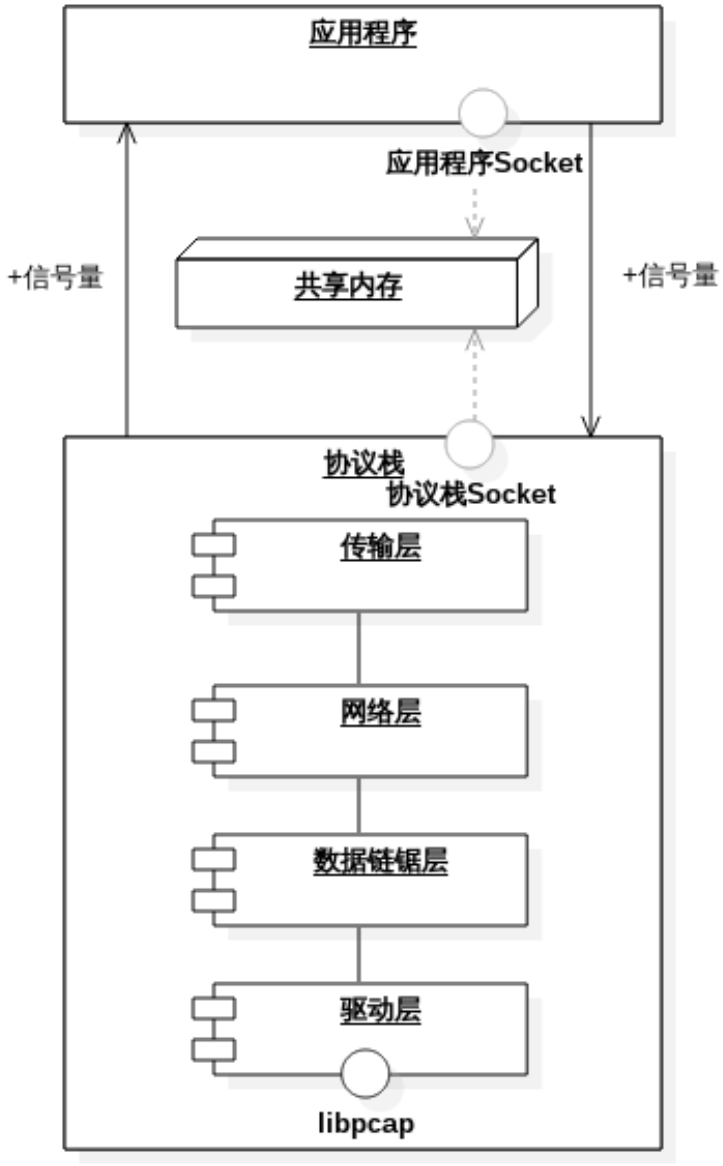
应用程序
应用程序Socket
+信号量
共享内存
+信号量
协议栈
协议栈Socket
传输层
网络层
数据链锯层
驱动层
libpcap

图 1

说 明 书 附 图

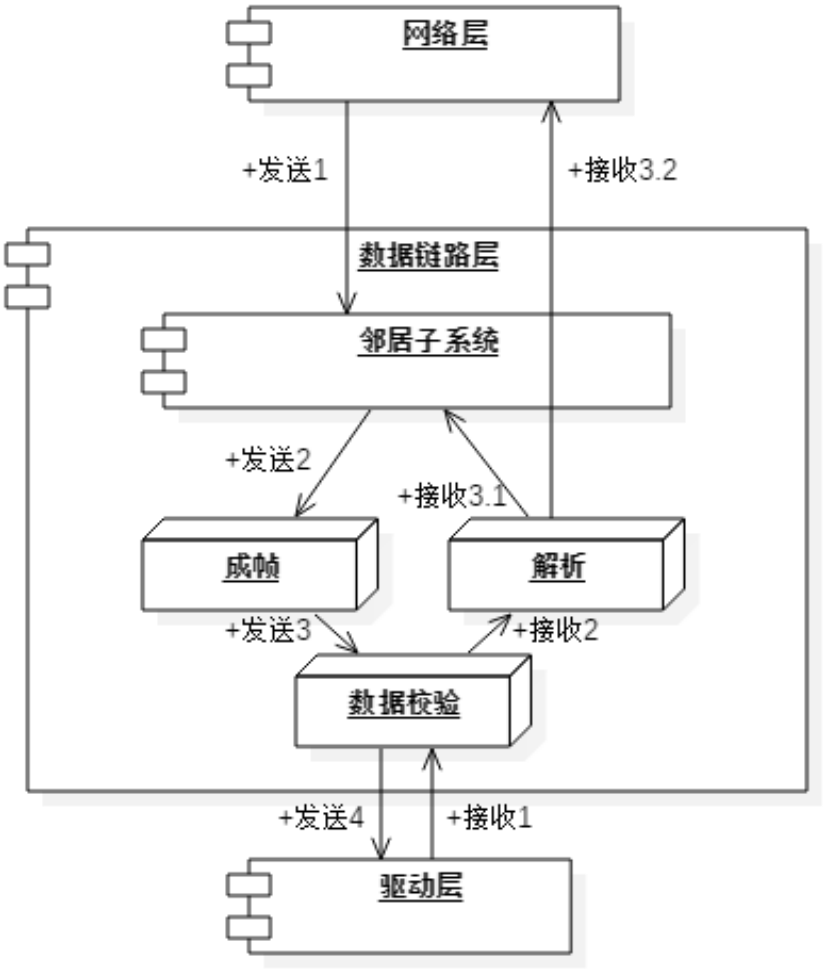
网络层
+发送1
+接收3.2
数据链路层
邻居子系统
+发送2
+接收3.1
成帧
解析
+发送3
+接收2
数据校验
+发送4
+接收1
驱动层

图 2

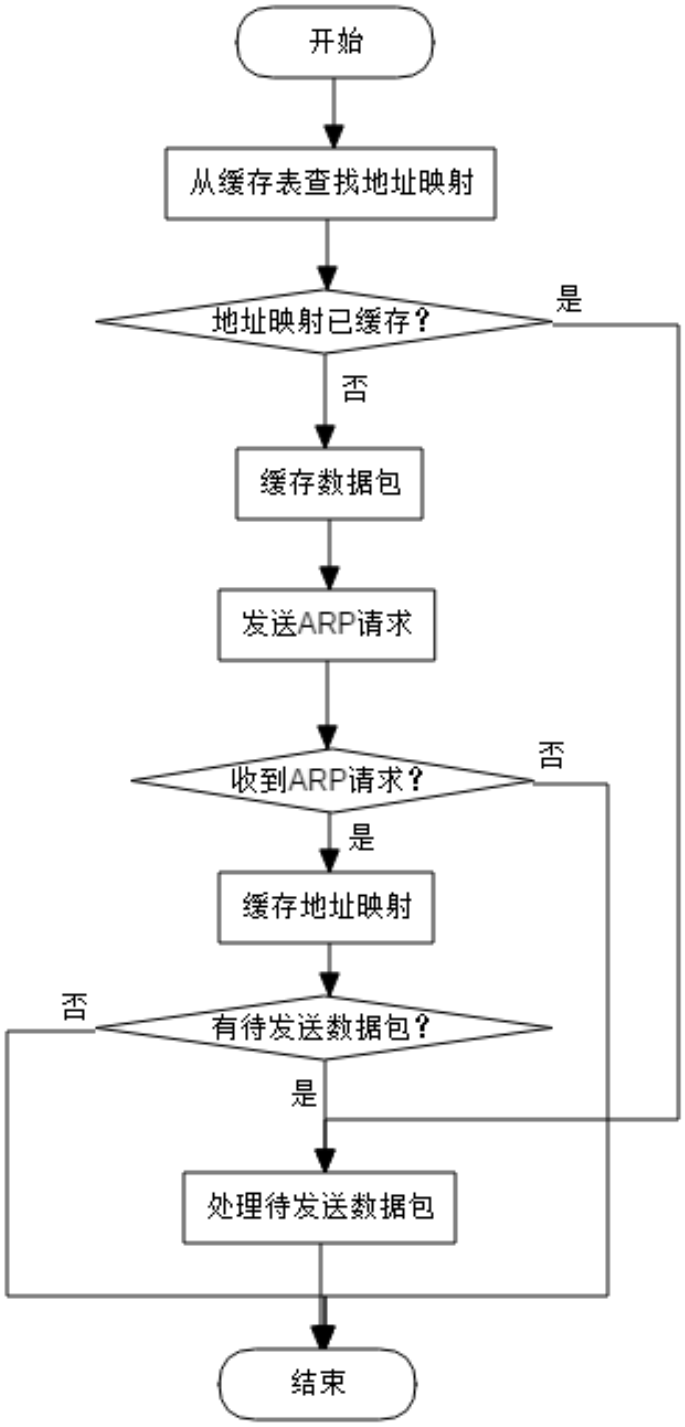
开始
从缓存表查找地址映射
地址映射已缓存？
是
否
缓存数据包
发送ARP请求
收到ARP请求？
否
是
缓存地址映射
有待发送数据包？
否
是
处理待发送数据包
结束

图 3

说 明 书 附 图

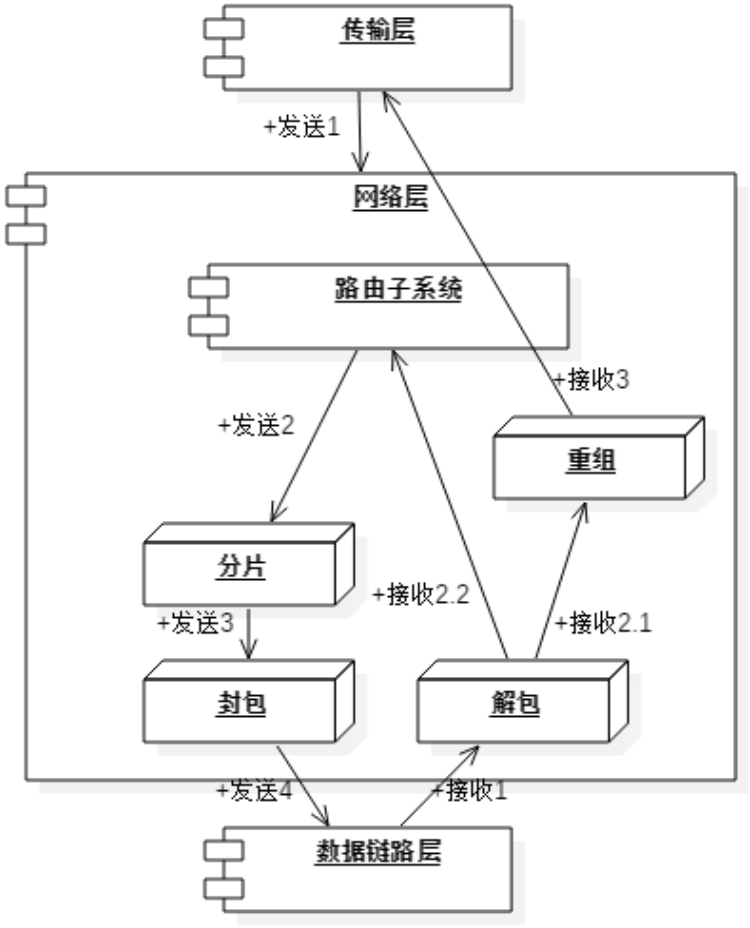
传输层
+发送1
网络层
路由子系统
+接收3
+发送2
重组
分片
+接收2.2
+发送3
+接收2.1
封包
解包
+发送4
+接收1
数据链路层

图 4

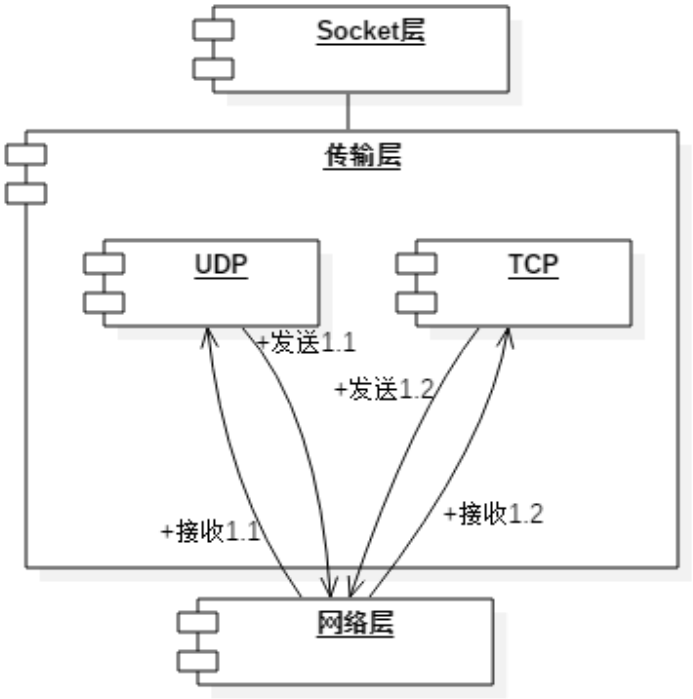
Socket层
传输层
UDP
TCP
+发送1.1
+发送1.2
+接收1.1
+接收1.2
网络层

图 5