



(12) 发明专利申请

(10) 申请公布号 CN 103997500 A

(43) 申请公布日 2014. 08. 20

(21) 申请号 201410244343. 2

(22) 申请日 2014. 06. 04

(71) 申请人 西北工业大学

地址 710072 陕西省西安市友谊西路 127 号

(72) 发明人 蒋泽军 杜承烈 陈进朝 黄云婷

侯亮

(74) 专利代理机构 西北工业大学专利中心

61204

代理人 陈星

(51) Int. Cl.

H04L 29/06 (2006. 01)

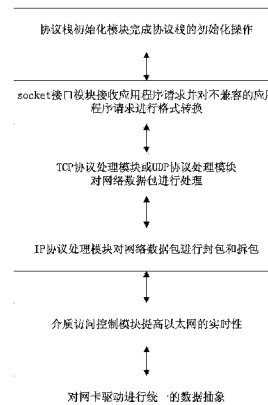
权利要求书1页 说明书4页 附图2页

(54) 发明名称

一种轻量级实时 TCP/IP 协议栈的实现方法

(57) 摘要

本发明提出一种轻量级实时 TCP/IP 协议栈的实现方法,通过初始化协议栈、socket 接口模块处理阶段、传输层处理阶段、网络层处理阶段、介质访问控制层处理阶段和网卡抽象层处理阶段对标准 TCP/IP 协议栈进行改进,增强了系统的可扩展性,提高了系统的实时性能,减少了系统开销和通信延迟。



1. 一种轻量级实时 TCP/IP 协议栈的实现方法,其特征在于:包括以下步骤:

步骤 1:初始化协议栈;

步骤 2:完成 socket 接口优先级、实时指标、阻塞方式、最大发送和接收缓冲区池大小的设置操作;完成 socket 接口的创建、绑定、监听、关闭以及数据收发接口实现;通过 socket 接口接收应用程序请求,并对不兼容的应用程序请求进行格式转换;

步骤 3:采用 TCP 协议处理模块或 UDP 协议处理模块对网络数据包进行处理;

步骤 4:采用 IP 协议处理模块对网络数据包进行封包和拆包;

步骤 5:在介质访问控制模块中采用基于 TDMA 的介质访问机制和令牌环方式得到所需的网络实时性;

步骤 6:在网卡驱动抽象层对网卡驱动进行统一的数据抽象;当网卡设备启用时,向该网卡驱动的数据抽象进行注册,并完成网卡驱动数据抽象与网卡驱动的连接;当需要发送的数据到达网卡驱动数据抽象层时,通过本地维护的路由表选择具体使用的网卡设备从而完成数据的发送。

2. 根据权利要求 1 所述一种轻量级实时 TCP/IP 协议栈的实现方法,其特征在于:初始化协议栈包括:完成网卡服务句柄的打开、网卡设备的绑定、打开操作;分配全局重发缓冲区池和全局发送缓冲区池;设置发送、接收线程优先级;增加系统服务表项;设置本机 IP 地址和 MAC 地址,并通过内存映射到用户空间。

3. 根据权利要求 1 或 2 所述一种轻量级实时 TCP/IP 协议栈的实现方法,其特征在于:创建 socket 接口时,设置 socket 接口的类型,在内核中为每个 socket 接口分配接收缓冲区池,并将之映射到用户空间;绑定 socket 接口时,完成 socket 接口、端口号和目的主机的绑定操作,在内核下向目的主机发送 ARP 请求以获得目的主机 MAC 地址,协议栈中采用静态 ARP 机制,完成地址解析。

4. 根据权利要求 3 所述一种轻量级实时 TCP/IP 协议栈的实现方法,其特征在于:socket 接口接收数据直接从缓冲区池中取出数据;socket 接口发送数据,直接写入到协议栈初始化时分配的全局发送缓冲区池中。

5. 根据权利要求 4 所述一种轻量级实时 TCP/IP 协议栈的实现方法,其特征在于:当网络采用 TCP 传输方式,socket 接口调用 TCP 协议处理模块进行数据包处理,所述 TCP 协议处理模块实现正常的 TCP 流程的状态机、对大数据包进行拆分,屏蔽 IP 分片功能、通过窗口大小完成流量和拥塞控制功能;当网络采用 UDP 传输方式,socket 接口调用 UDP 协议处理模块进行数据包处理,所述 UDP 协议处理模块完成数据包的 UDP 协议封装、通过对大数据包进行拆分,屏蔽 IP 分片功能,省略 UDP 校验和计算功能。

6. 根据权利要求 5 所述一种轻量级实时 TCP/IP 协议栈的实现方法,其特征在于:IP 协议处理模块实现 IP 数据包的封包和拆包,屏蔽 IP 分片,将 TOS 字段和分片偏移字段作为预留字段;其中 IP 协议处理模块包括 ARP 协议处理模块和 ICMP 协议处理模块,所述 ARP 协议处理模块将 IP 地址映射为 MAC 地址;所述 ICMP 协议处理模块对网络控制报文进行发送和解析。

7. 根据权利要求 6 所述一种轻量级实时 TCP/IP 协议栈的实现方法,其特征在于:基于 TDMA 的介质访问机制对各个节点的数据帧发送时刻进行控制;采用令牌环方式发送数据,任意终端站在传输之前能够计算出最大等待时间。

一种轻量级实时 TCP/IP 协议栈的实现方法

技术领域

[0001] 本发明属于计算机网络和数据通信技术领域,具体为一种轻量级实时 TCP/IP 协议栈的实现方法。

背景技术

[0002] 目前标准 TCP/IP 协议栈其实现复杂,且不能满足特定应用对实时性的要求,而为保证通信实时性的解决方案,大多是对数据链路层进行更改和扩展,诸如以下三种方案:

[0003] 1. 由奥地利贝加莱公司提出的 Ethernet Powerlink,引入了 SCNM(时间槽通信网络管理)算法来保证实时以太网通信的确定性。

[0004] 2. 由德国倍福公司提出的 EtherCAT(Ethernet for Control Automation Technology),其采用 IEEE1588 标准中定义的精确时钟同步机制,引入了时间戳数据类型座位扩展。

[0005] 3. 我国第一个拥有自主知识产权的现场总线国家标准:《EPA 通信标准》对 ISO/IEC8802.3 协议规定的链路层进行扩展,增加了通信调度管理实体(Communication Scheduling Management Entity, CSME)。

[0006] 但这些方案仅为了保证通信实时性,而对于协议栈实现复杂的缺点仍未得到解决,因此提出一种轻量级实时对 TCP/IP 协议栈的实现方法具有重要意义。

发明内容

[0007] 本发明旨在实现一种轻量级实时 TCP/IP 协议栈,使之不仅实现简单,使用方便,减少系统开销和通信延迟,且能满足特定应用对实时性能的要求。

[0008] 本发明的技术方案为:

[0009] 所述一种轻量级实时 TCP/IP 协议栈的实现方法,其特征在于:包括以下步骤:

[0010] 步骤 1:初始化协议栈;

[0011] 步骤 2:完成 socket 接口优先级、实时指标、阻塞方式、最大发送和接收缓冲区池大小的设置操作;完成 socket 接口的创建、绑定、监听、关闭以及数据收发接口实现;通过 socket 接口接收应用程序请求,并对不兼容的应用程序请求进行格式转换;

[0012] 步骤 3:采用 TCP 协议处理模块或 UDP 协议处理模块对网络数据包进行处理;

[0013] 步骤 4:采用 IP 协议处理模块对网络数据包进行封包和拆包;

[0014] 步骤 5:在介质访问控制模块中采用基于 TDMA 的介质访问机制和令牌环方式得到所需的网络实时性;

[0015] 步骤 6:在网卡驱动抽象层对网卡驱动进行统一的数据抽象;当网卡设备启用时,向该网卡驱动的数据抽象进行注册,并完成网卡驱动数据抽象与网卡驱动的连接;当需要发送的数据到达网卡驱动数据抽象层时,通过本地维护的路由表选择具体使用的网卡设备从而完成数据的发送。

[0016] 进一步的优选方案,所述一种轻量级实时 TCP/IP 协议栈的实现方法,其特征在

于:初始化协议栈包括:完成网卡服务句柄的打开、网卡设备的绑定、打开操作;分配全局重发缓冲区池和全局发送缓冲区池;设置发送、接收线程优先级;增加系统服务表项;设置本机 IP 地址和 MAC 地址,并通过内存映射到用户空间。

[0017] 进一步的优选方案,所述一种轻量级实时 TCP/IP 协议栈的实现方法,其特征在于:创建 socket 接口时,设置 socket 接口的类型,在内核中为每个 socket 接口分配接收缓冲区池,并将之映射到用户空间;绑定 socket 接口时,完成 socket 接口、端口号和目的主机的绑定操作,在内核下向目的主机发送 ARP 请求以获得目的主机 MAC 地址,协议栈中采用静态 ARP 机制,完成地址解析。

[0018] 进一步的优选方案,所述一种轻量级实时 TCP/IP 协议栈的实现方法,其特征在于:socket 接口接收数据直接从缓冲区池中取出数据;socket 接口发送数据,直接写入到协议栈初始化时分配的全局发送缓冲区池中。

[0019] 进一步的优选方案,所述一种轻量级实时 TCP/IP 协议栈的实现方法,其特征在于:当网络采用 TCP 传输方式,socket 接口调用 TCP 协议处理模块进行数据包处理,所述 TCP 协议处理模块实现正常的 TCP 流程的状态机、对大数据包进行拆分,屏蔽 IP 分片功能、通过窗口大小完成流量和拥塞控制功能;当网络采用 UDP 传输方式,socket 接口调用 UDP 协议处理模块进行数据包处理,所述 UDP 协议处理模块完成数据包的 UDP 协议封装、通过对大数据包进行拆分,屏蔽 IP 分片功能,省略 UDP 校验和计算功能。

[0020] 进一步的优选方案,所述一种轻量级实时 TCP/IP 协议栈的实现方法,其特征在于:IP 协议处理模块实现 IP 数据包的封包和拆包,屏蔽 IP 分片,将 TOS 字段和分片偏移字段作为预留字段;其中 IP 协议处理模块包括 ARP 协议处理模块和 ICMP 协议处理模块,所述 ARP 协议处理模块将 IP 地址映射为 MAC 地址;所述 ICMP 协议处理模块对网络控制报文进行发送和解析。

[0021] 进一步的优选方案,所述一种轻量级实时 TCP/IP 协议栈的实现方法,其特征在于:基于 TDMA 的介质访问机制对各个节点的数据帧发送时刻进行控制;采用令牌环方式发送数据,任意终端站在传输之前能够计算出最大等待时间。

[0022] 有益效果

[0023] 本发明的创新点和优点如下:

[0024] 1) 在基于介质访问控制模块中,采用两种方式来解决通信延迟:基于 TDMA 的介质访问机制对各个节点的数据帧发送时刻进行控制;采用令牌环方式发送数据,任意终端站在传输之前可以计算出最大等待时间。通过这两种方式,来得到所需的网络实时性。

[0025] 2) 设置线程和数据包优先级:协议栈初始化阶段,设置发送和接收线程优先级,使协议栈能够高效地发送和接收数据,避免了其他线程的抢占,影响发送和接收数据的速率;socket 接口模块设置数据包的优先级,保证优先级高的数据能够优先发送。

[0026] 3) 整个数据发送和接收的过程,通过内存映射技术,将内核空间映射到用户空间,同时采用零拷贝技术减少数据包穿越 TCP/IP 协议栈时的复制次数,以提高以太网的实时性能。

[0027] 4) 通过对标准 TCP/IP 协议栈进行裁剪和优化,使 TCP/IP 协议栈实现简单,使用方便,减少系统开销和网络延迟。TCP 协议处理模块,对大数据包进行拆分,屏蔽 IP 分片功能、通过窗口大小来完成流量和拥塞控制;IP 协议处理模块屏蔽 IP 分片,将 TOS 字段和分片偏

移字段作为预留字段。ARP 协议模块将 IP 地址映射为 MAC 地址,采用静态 ARP 机制,在网络初始化时完成地址解析。

[0028] 5) 提供不同协议栈之间的兼容性,以提供一套统一的 API 接口给应用层。

附图说明

[0029] 图 1 为本发明实时 TCP/IP 协议栈的实现方法流程图;

[0030] 图 2 为本发明协议组织架构示意图。

具体实施方式

[0031] 下面结合具体实施例描述本发明:

[0032] 本发明的实时 TCP/IP 协议栈的实现方法对标准 TCP/IP 协议栈进行改进,增强了系统的可扩展性,提高了系统的实时性能,减少了系统开销和通信延迟。

[0033] 在本实施方式中,首先本发明的实时 TCP/IP 协议栈包括 IP 协议, TCP 协议, UDP 协议, ICMP 协议和 ARP 协议;其次,为了增强实时性能,增加了介质访问控制模块;最后,为了增强可扩展性和应用程序的兼容性,增加了 socket 接口模块和网卡驱动抽象层;其中介质访问控制模块是为了满足网络通信的实时要求;socket 接口模块是为了符合通用的网络 socket 编程接口,兼容大多数的网络应用程序;网卡驱动抽象层是对网卡驱动的统一管理层,在实时 TCP/IP 协议栈中进行处理后的数据将通过网卡驱动抽象层发送到具体的网卡设备上,具体步骤如下:

[0034] 步骤 1:初始化协议栈:完成网卡服务句柄的打开、网卡设备的绑定、打开操作;分配全局重发缓冲区池和全局发送缓冲区池,使得数据发送过程中无需进行内存分配;设置发送、接收线程优先级,使协议栈能够高效地发送和接收数据,进而避免其他线程的抢占;增加系统服务表项,以完成系统服务函数的添加;设置本机 IP 地址和 MAC 地址,并通过内存映射到用户空间;用户发送数据时,可以将数据直接写入该缓冲区池,从而省略了用户到内核的数据拷贝过程。

[0035] 步骤 2:socket 接口模块处理阶段:完成 socket 接口优先级、实时指标、阻塞方式、最大发送和接收缓冲区池大小的设置操作;并对不兼容的应用请求进行格式转换,保证优先级高的数据优先发送,提高以太网的实时性能;完成 socket 的创建、绑定、监听、关闭以及数据收发接口实现。

[0036] 其中创建 socket 时,设置 socket 的类型,在内核中为每个 socket 分配接收缓冲区池,并将之映射到用户空间;绑定 socket 时,完成 socket、端口号和目的主机的绑定操作,在内核下向目的主机发送 ARP 请求以获得目的主机 MAC 地址,协议栈中采用静态 ARP 机制,完成地址解析,使得数据发送过程中无需再次发送 ARP 请求。

[0037] 收发数据,通过内存映射和零拷贝技术,使之具有更高的效率,其中,接收数据直接从缓冲区池中取出数据,减少了内核到用户的数据拷贝过程;发送数据,直接写入到协议栈初始化时分配的全局发送缓冲区池中,减少了用户到内核的数据拷贝操作。具体实施过程中,应用程序通过调用 socket 接口完成网络编程,socket 接口模块根据应用程序调用的 socket 接口函数完成具体的操作。在本实施方案中,socket 接口模块只处理 TCP/IP 协议族的请求,其他协议族的请求不支持,且只支持以 SOCK_DGRAM 和 SOCK_STREAM 为参数的套

接字传输类型。

[0038] 步骤3:传输层处理阶段:采用TCP协议处理模块或UDP协议处理模块对网络数据包进行处理;当网络采用TCP传输方式后,socket将调用TCP协议处理模块进行数据包处理,所述TCP协议处理模块会实现正常的TCP流程的状态机、对大数据包进行拆分,屏蔽IP分片功能、通过窗口大小来完成流量和拥塞控制等功能;当网络采用UDP传输方式后,socket将调用UDP协议处理模块进行数据包处理,所述UDP协议处理模块会完成数据包的UDP协议封装、通过对大数据包进行拆分,屏蔽IP分片功能,省略UDP校验和的计算等功能。

[0039] 步骤4:网络层处理阶段:采用IP协议处理模块对网络数据包进行封包和拆包;IP协议处理模块实现了IP数据包的封包和拆包,屏蔽IP分片,将TOS字段和分片偏移字段作为预留字段。IP协议处理模块包括ARP协议处理模块和ICMP协议处理模块,所述ARP协议处理模块将IP地址映射为MAC地址;所述ICMP协议处理模块对网络控制报文进行发送和解析。

[0040] 步骤5:介质访问控制层处理阶段:采用两种方式来解决通信延迟:采用基于TDMA的介质访问机制对各个节点的数据帧发送时刻进行控制;采用令牌环方式发送数据,任意终端站在传输之前可以计算出最大等待时间。通过这两种方式,来得到所需的网络实时性。

[0041] 步骤6:网卡抽象层处理阶段:在网卡驱动抽象层对网卡驱动进行统一的数据抽象,由于网络设备所采用的网卡多且网卡驱动差异很大,所以通过驱动的数据抽象来规范网卡驱动的统一化管理。

[0042] 这一步主要维护当前可用网卡设备,当网卡设备启用时,向该网卡驱动的数据抽象进行注册,并完成网卡驱动数据抽象与网卡驱动的连接;当需要发送的数据到达网卡驱动数据抽象层时,通过本地维护的路由表选择具体使用的网卡设备从而完成数据的发送。

[0043] 本发明的实时TCP/IP协议栈的实现方法对标准TCP/IP协议栈进行改进,增强了系统的可扩展性,提高了系统的实时性能,减少了系统开销和通信延迟。

[0044] 最后应说明的是:以上实施例是仅用以说明而非限制本发明的技术方案,尽管参照上述实施例对本发明进行了详细说明,本领域的技术人员应当理解:您依然可以对本发明进行修改或者等同替换,并且不脱离本发明的精神和范围的任何修改或局部替换,其均应涵盖在本发明的权利要求范围当中。

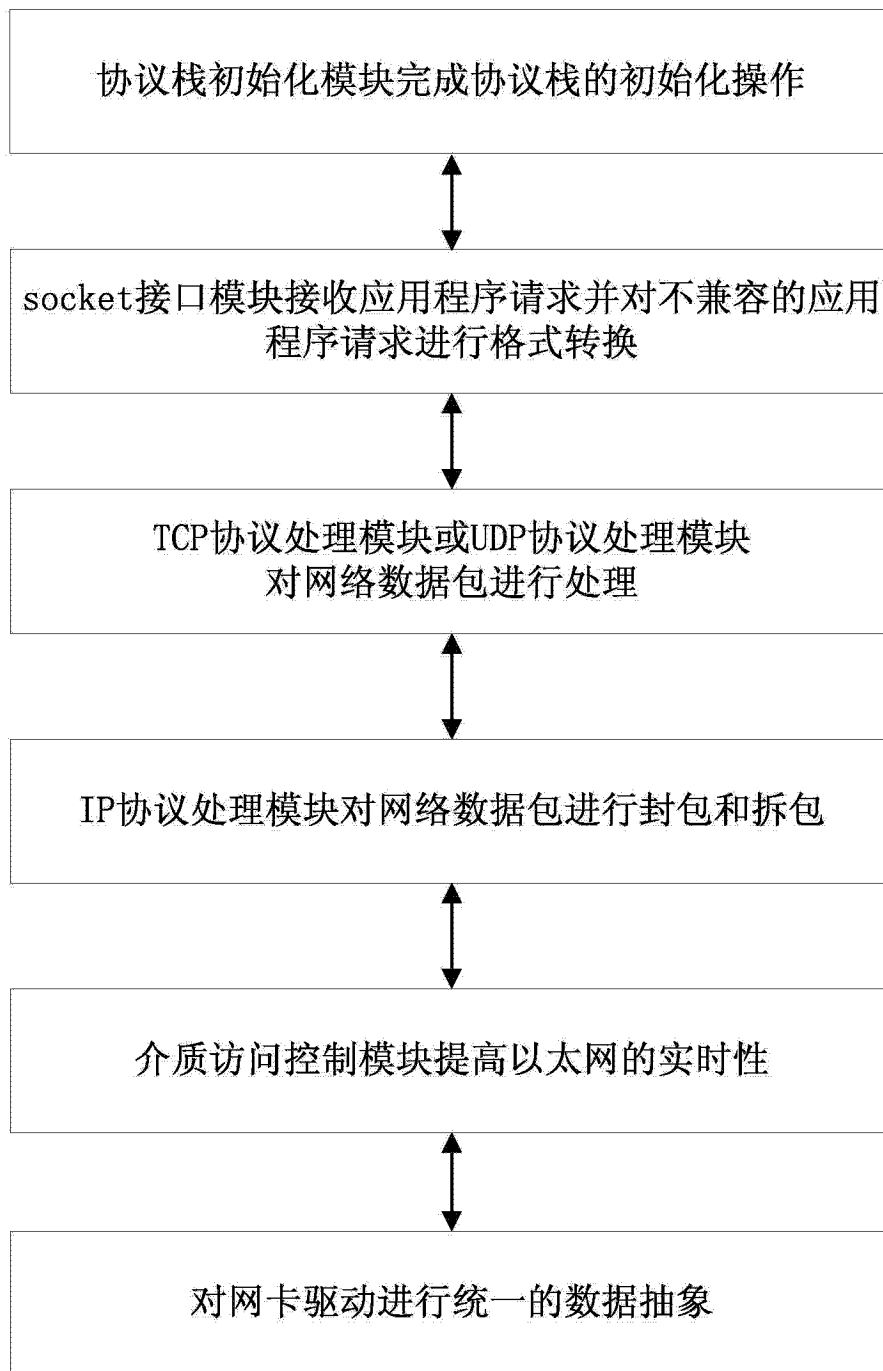


图 1

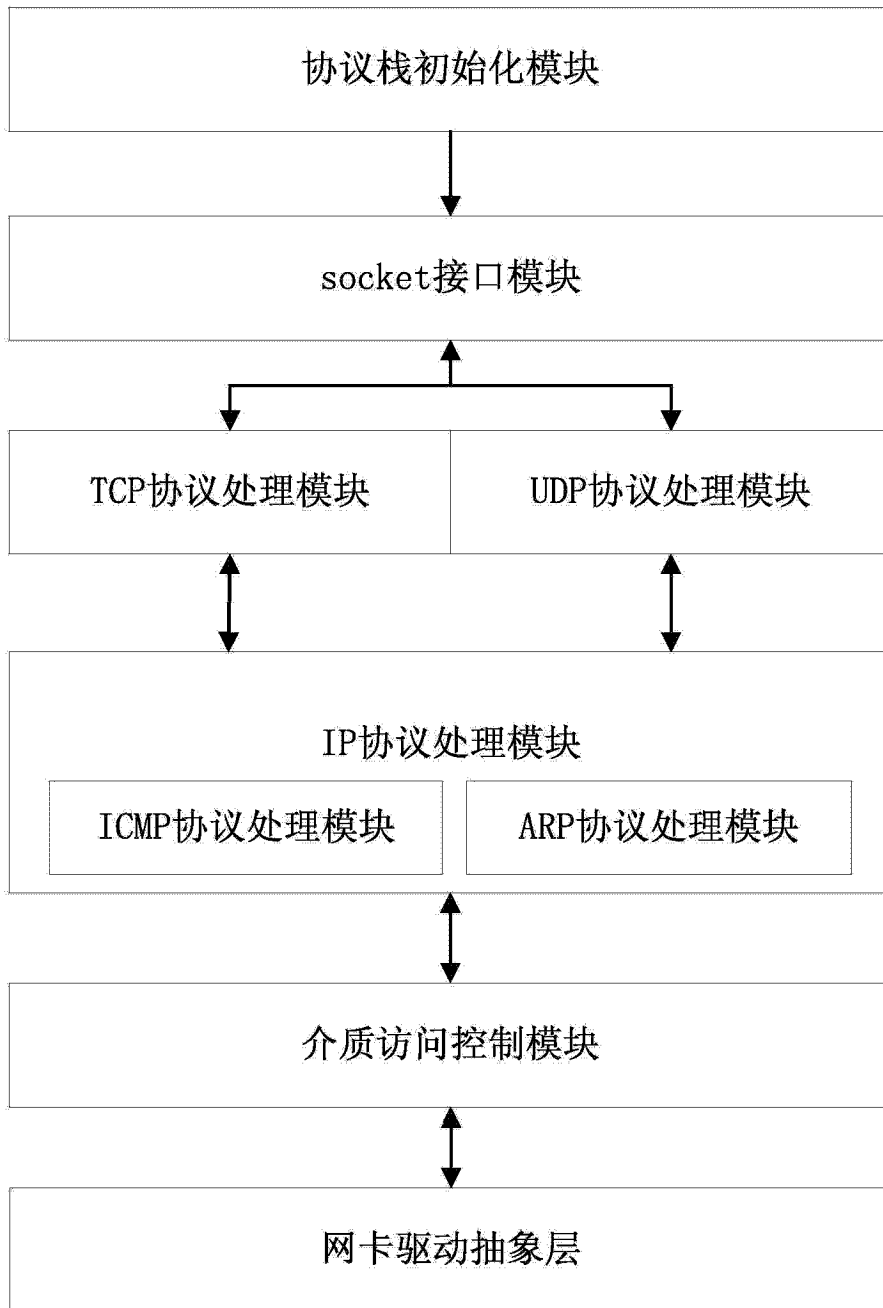


图 2