(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101263686 B
(45)授权公告日 2014.11.12

(21)申请号 200680033111.6
(22)申请日 2006.09.11
(30)优先权数据
60/716,122 2005.09.12 US
11/275,185 2005.12.16 US
(85)PCT国际申请进入国家阶段日
2008.03.10
(86)PCT国际申请的申请数据
PCT/US2006/035497 2006.09.11
(87)PCT国际申请的公布数据
WO2007/033179 EN 2007.03.22
(73)专利权人 微软公司
地址 美国华盛顿州
(72)发明人 M·T·玛萨 D·A·迪昂
R·欧帕弗斯基
(74)专利代理机构 上海专利商标事务所有限公司 31100
代理人 陈斌
(51)Int.Cl.
H04L 12/28(2006.01)
H04L 29/10(2006.01)
G06F 15/16(2006.01)
(56)对比文件
US 20040008717 A1,2004.01.15, 说明书第0030,0079-0081段、附图1,5-6.
US 20040078625 A1,2004.04.22, 全文.
CN 1404671 A,2003.03.19, 全文.
审查员 刘金凤

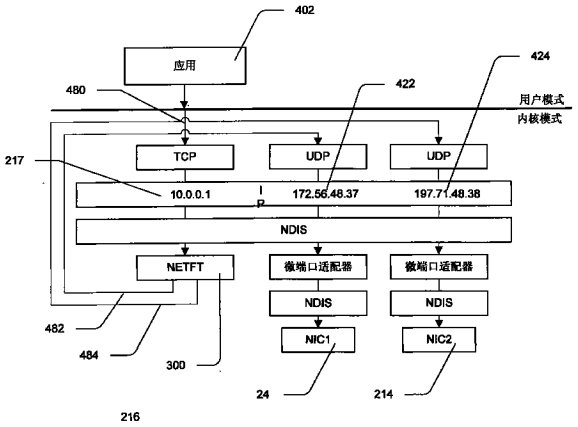
权利要求书3页 说明书11页 附图11页

(54)发明名称

用于向应用提供多个节点之间的容错网络通信的方法和系统

(57)摘要

一种用于向应用提供多个节点之间的容错网络通信的方法,包括:通过在多个节点之间耦合的多个网络上提供多个初始通信通路;从应用接收发送节点上的数据分组,该发送节点该多个节点中的一个,数据分组由应用添加多个节点中的一个上的地址;以及在多个初始通信通路中为该数据分组选择第一所选通路,其中该第一所选通路是优选通路。



CN 101263686 B

1. 一种用于向应用提供多个节点之间的容错网络通信的方法，包括：

经由耦合所述多个节点的多个网络提供多个初始通信通路，每个网络独立于其它网络；

在运行在发送节点上的多个协议驱动器中的第二协议驱动器上从运行在发送节点上的所述应用接收数据分组，所述发送节点为所述多个节点之一，所述数据分组由所述应用添加所述多个节点中的除所述发送节点之外的一个节点的地址；以及

由运行在发送节点上的容错通信驱动器在所述多个初始通信通路中为所述数据分组选择第一所选通路，其中所述第一所选通路是优选通路，所述优选通路是由所述容错通信驱动器基于优先级信息计算的，其中，所述发送节点包括多个网络接口卡，每个网络接口卡耦合到所述多个网络中不同的一个，所述多个网络接口卡中的第一网络接口卡由耦合到第一协议驱动器的第一网络接口卡驱动器控制，所述第一协议驱动器耦合到所述容错通信驱动器，所述容错通信驱动器耦合到所述第二协议驱动器，所述第二协议驱动器耦合到所述应用。

2. 如权利要求 1 所述的方法，其特征在于，还包括：

检测与所述多个初始通信通路中的一个相关联的本地连通性事件；以及

基于所述本地连通性事件指示所述多个初始通信通路中的所述一个是否可用。

3. 如权利要求 1 所述的方法，其特征在于，还包括：

在所述多个初始通信通路中的一个上发送可路由心跳分组；

监视对所述可路由心跳分组的响应以确定所述多个初始通信通路中的所述一个的端对端操作状态；以及

基于对所述响应的监视指示所述多个初始通信通路中的所述一个是否是端对端可用的。

4. 如权利要求 1 所述的方法，其特征在于，所述地址是网际协议第 4 版的地址或网际协议第 6 版的地址。

5. 如权利要求 1 所述的方法，其特征在于，从所述应用接收到的所述数据分组是传输控制协议分组或用户数据报协议分组。

6. 如权利要求 1 所述的方法，其特征在于，用于执行所述方法的计算机可执行指令被存储在计算机可读介质上。

7. 如权利要求 1 所述的方法，其特征在于，所述地址是虚拟地址。

8. 如权利要求 7 所述的方法，其特征在于，还包括将所述数据分组在所述第一所选通路上可路由地隧穿而无需所述应用了解所述多个初始通信通路的哪一个是所选通路。

9. 如权利要求 8 所述的方法，其特征在于，在所述第一所选通路上可路由地隧穿的所述数据分组是传输控制协议分组或用户数据报协议分组。

10. 如权利要求 7 所述的方法，其特征在于，还包括：

检测所述第一所选通路的故障；

在所述多个初始通信通路中选择第二所选通路，其中所述第二所选通路是端对端可用的且是优选通路；以及

将所述数据分组在所述第二所选通路上可路由地隧穿而无需所述应用了解所述多个初始通信通路的哪一个是所述第二所选通路。

11. 如权利要求 1 所述的方法，其特征在于，所述第一所选通路是基于包括在所述数据分组中的物理目的地址来选择的。

12. 如权利要求 11 所述的方法，其特征在于，还包括在所述第一所选通路上发送所述数据分组而无需所述应用了解所述多个初始通信通路的哪一个是所述第一所选通路。

13. 如权利要求 11 所述的方法，其特征在于，还包括 ：

检测所述第一所选通路的故障 ；

从所述多个初始通信通路选择第二所选通路，其中所述第二所选通路是端对端可用的且是优选通路 ；以及

将所述数据分组在所述第二所选通路上可路由地隧穿而无需所述应用了解所述多个初始通信通路的哪一个是所述第二所选通路。

14. 如权利要求 13 所述的方法，其特征在于，在所述第二所选通路上可路由地隧穿的所述数据分组是传输控制协议分组或用户数据报协议分组。

15. 一种用于向应用提供多个节点之间的容错网络通信的方法，包括 ：

经由耦合所述多个节点的多个网络提供多个初始通信通路，每个网络独立于其它网络 ；

在运行在接收节点上的多个协议驱动器中的第一协议驱动器上接收数据分组，所述接收节点包括容错通信驱动器并且是所述多个节点之一，所述数据分组指向所述应用，所述应用运行在所述接收节点上，所述数据分组是从所述多个节点中的发送节点发送的，所述发送节点包括相应的容错通信驱动器且所述数据分组是经该相应的容错通信驱动器发送的 ；

确定所述数据分组是否被可路由地隧穿 ；

如果所述数据分组被可路由地隧穿，则由所述第一协议驱动器将所述数据分组转发到所述容错通信驱动器 ；以及

由所述容错通信驱动器将所述数据分组提供给所述多个协议驱动器中的第二协议驱动器，

其中，所述接收节点包括多个网络接口卡，每个网络接口卡耦合到所述多个网络中不同的一个，所述多个网络接口卡中的第一网络接口卡由耦合到第一协议驱动器的第一网络接口卡驱动器控制，所述第一协议驱动器耦合到所述容错通信驱动器，所述容错通信驱动器耦合到第二协议驱动器，所述第二协议驱动器耦合到所述应用。

16. 如权利要求 15 所述的方法，其特征在于，还包括 ：

通过所述多个初始通信通路之一在所述接收节点上接收可路由心跳分组 ；以及

以对所述可路由心跳分组的响应进行回复以指示所述多个初始通信通路的所述之一的端对端操作状态。

17. 如权利要求 15 所述的方法，其特征在于，所述数据分组是传输控制协议分组或用户数据报协议分组。

18. 如权利要求 15 所述的方法，其特征在于，用于执行所述方法的计算机可执行指令被存储在计算机可读介质上。

19. 一种用于向应用提供多个节点之间的容错网络通信的系统，包括 ：

第一容错通信驱动器，耦合到第一网络堆栈并且在第一节点上运行，所述第一节点是

所述多个节点中的一个，

所述第一容错通信驱动器耦合到第二容错通信驱动器，所述第二容错通信驱动器在所述多个节点中的第二节点上运行，所述第一容错通信驱动器和所述第二容错通信驱动器经由耦合到所述多个节点的多个网络上的多个初始通信通路来耦合，其中每个网络独立于其它网络，其中，所述第一节点包括多个网络接口卡，每个网络接口卡耦合到所述多个网络中不同的一个，所述多个网络接口卡中的第一网络接口卡由耦合到第一协议驱动器的第一网络接口卡驱动器控制，所述第一协议驱动器耦合到所述第一容错通信驱动器，所述第一容错通信驱动器耦合到第二协议驱动器，所述第二协议驱动器耦合到所述应用。

20. 如权利要求 19 所述的系统，其特征在于，所述第一容错通信驱动器包括：

微端口适配器，经由所述第一网络堆栈耦合到所述应用；

路由数据库，耦合到所述微端口适配器，所述路由数据库包括：

表示从所述第一节点到所述第二节点的通路的条目，包括所述第二节点的物理地址，所述通路是所述多个初始通信通路中的一个；以及

对从所述第一节点到所述第二节点的所述通路的端对端操作状态的指示；

协议适配器，耦合到所述微端口适配器并经由运行在所述第一节点上的第一网络接口卡耦合到所述多个网络中的一个，所述多个网络中的所述一个提供从所述第一节点到所述第二节点的所述通路；以及

隧穿适配器，耦合到所述微端口适配器以及第三协议驱动器，所述第三协议驱动器耦合到运行在所述第一节点上的第二网络接口卡。

用于向应用提供多个节点之间的容错网络通信的方法和系统

[0001] 背景

[0002] 在计算机联网环境中，多个节点可在网络上彼此通信。如果网络发生故障，则这些节点之间的通信可能中断。

[0003] 概述

[0004] 以下呈现了本公开的简化概述以便向读者提供基本的了解。本概述并非本公开的宽泛综述，也并不标识本发明的关键或重要要素或者刻划本发明的范围。其目的仅是为了以简化方式呈现在此所公开的某些概念作为在随后呈现的更详细描述的序言。

[0005] 以下示例通过在网络化节点上运行的应用软件经由要求最小考虑的独特的网络堆栈体系结构来提供计算机网络通信容错。

[0006] 许多附带特征将变得更容易理解，因为通过结合附图参考以下详细描述能获得更好的理解。

[0007] 附图描述

[0008] 根据附图阅读以下详细描述将更好地理解本发明，其中：

[0009] 图 1 是示出了示例网络堆栈体系结构的框图。

[0010] 图 2 是示出了包括经由两个网络耦合的两个示例节点的网络化计算环境的框图。

[0011] 图 3 是示出了示例容错通信驱动器 NETFT 的框图。

[0012] 图 4 是示出了包括 NETFT 和应用的示例容错通信体系结构的框图。

[0013] 图 5 是示出了数据流过包括经由网络 1 上的路径 A 和网络 2 上的路径 B 耦合的源节点和目的节点的容错通信环境的流程图。

[0014] 图 6 是示出了在外加若干可能的通信故障的情况下数据流过图 5 所示的容错通信环境的流程图。

[0015] 图 7 是示出了容错通信驱动器 NETFT 的另一示例的框图。

[0016] 图 8 是示出了包括 NETFT 和应用的示例容错通信体系结构的框图。

[0017] 图 9 是示出了数据流过包括经由网络 1 上的路径 A 和网络 2 上的路径 B 耦合的源节点和目的节点的容错通信环境的流程图。

[0018] 图 10 是示出了在外加若干可能的通信故障的情况下数据流过图 9 所示的容错通信环境的流程图。

[0019] 图 11 是示出了其中可实现上述技术的示例计算环境的框图。

[0020] 附图中，类似的附图标记用于指示类似的部分。

[0021] 详细描述

[0022] 以下结合附图提供的详细描述旨在作为本发明的示例的描述而非旨在表示其中可构建或使用本发明的示例的唯一形式。此描述阐述了这些示例的功能以及用于构建和操作这些示例的步骤序列。然而，相同或等效的功能以及序列可通过不同的示例来实现。

[0023] 尽管本发明的示例在此被描述并示为在计算和联网系统中实现，但是所述系统作为示例而非限制而提供。如本领域技术人员理解的，本发明的示例适于应用在各种不同类

型的计算和联网环境中。

[0024] 图 1 是示出了示例网络堆栈体系结构 100 的框图。网络堆栈（“堆栈”）通常经由网络堆栈接口和 / 或其它接口与软件应用程序耦合以向应用提供网络通信功能。应用通常被认为是处于（或耦合到）堆栈的“顶部”。网络通常被认为是处于（或耦合到）堆栈的“底部”。网络堆栈的各个要素可称为处于或靠近堆栈的顶部或底部，或者在该堆栈中相对彼此的较高或较低处。例如，在图 1 中，协议驱动器 130 在堆栈中高于在此特定附图中处于堆栈的底部的 NIC 180。如本领域技术人员所理解的，取决于描述的目的或焦点，网络堆栈的各种描述可包括或不包括某些堆栈要素，或者可以各种方式对这些要素进行分组、排序或命名。

[0025] 本文所用的术语“驱动器”称为控制程序等，它们使节点能够与诸如打印机、网络接口卡或其它计算机子系统的特定设备一起操作、或与诸如网络堆栈、协议驱动器和 / 或其它计算机软件或固件等一起操作。例如，协议驱动器通常与网络堆栈一起操作。

[0026] 应用可将数据分组传递到在另一节点上运行的应用所指定的堆栈。在此情况中，数据被认为是沿堆栈“向下”流动，并在网络上发送。由节点接收到的数据被认为是沿堆栈“向上”流动直至其到达所指定的应用。这些网络化系统对于本领域 技术人员是众所周知的。

[0027] 在一个示例中，堆栈是基于网络驱动器接口规范（“NDIS”），该规范定义了诸如 NIC 180 的网络接口卡（“NIC”）的标准应用程序设计接口（“API”），并从网络驱动器提取网络硬件。NDIS 也指定了分层网络驱动器之间的标准网络接口，由此从诸如协议驱动器的上层驱动器提取诸如微端口的、管理硬件的下层驱动器。多个遵循 NDIS 的协议驱动器可在单个节点上共存。而且，如果节点可能由于连接到多个网络而包括多个 NIC，则 NDIS 经由通信量所指示的其相关联驱动器将网络通信量路由至适当的 NIC。图 1 中示出了 NDIS 的示图。诸如开放数据链路接口（“ODI”）、数据链路提供者接口（“DLPI”）、统一驱动器接口（“UDI”）或其它技术的其它联网堆栈标准、技术和 / 或体系结构可在以下的示例中使用，并可进行适当的修改，如本领域技术人员应当理解的。为了方便起见，在此整个描述中，示例中使用 NDIS 和 NDIS 技术，但是其它标准、技术和 / 或体系结构可通过进行适当的修改而在所有这些示例中使用，除非另外指明。

[0028] 如图 1 中所示，经由 NDIS 120 耦合到 NIC 180 的是微端口驱动器 160。微端口驱动器通常经由 NDIS 微端口接口 162 与 NDIS 交互。微端口驱动器 160 可与 NIC 180 相关联，并且可管理其操作，包括通过 NIC 发送或接收数据。微端口驱动器 160 通常与诸如中间驱动器 140 和协议驱动器 130 的上层驱动器接口。微端口驱动器被视为 NIC 驱动器。NIC 微端口通常执行使用由 NDIS 所提供的通用或与 NIC 无关的功能来管理特定 NIC 所需的这些硬件专用操作。节点可包括多个 NIC，并且每个 NIC 通常具有相关联的 NIC 驱动器。本描述中的某些示例描述了微端口驱动器的使用，但是如本领域技术人员应当理解的，任何类型的 NIC 驱动器等可用在这些示例中，除非另外指明。

[0029] 协议或传输驱动器 130 经由 NDIS 协议接口 134 耦合到 NDIS 120。协议驱动器或传送协议驱动器通常提供用以创建、发送和接收数据分组的功能，这些数据分组通过网络堆栈或在网络上从一个节点发往另一节点。如本领域技术人员所公知的，通用的可靠或有保证的递送传输协议可以是 TCP/IP（传输控制协议 / 网际协议）。IP 上的 UDP（用户数据

报协议）可以是普通的非可靠或无保证的递送协议。诸如 IPX/SPX（因特网分组交换 / 顺序分组交换）的 TCP、UDP 和 / 或其它协议可在以下的示例中使用，除非另外指明。

[0030] 图 1 中示出了在协议驱动器 130 与 NDIS NIC 微端口 160 之间的 NDIS 中间（“IM”）驱动器 140。对于协议驱动器，IM 驱动器如同是 NDIS 微端口，而对 于 NIC 驱动器，它们如同是协议驱动器。沿网络堆栈向上或向下流动的数据分组经过 IM 驱动器 140，该驱动器 140 可忽略、检查、过滤、转发、重定向和 / 或更改数据分组。中间驱动器 140 也可称为过滤驱动器。

[0031] 图 2 是示出了包括经由两个网络 202 和 282 耦合的两个示例节点 210 和 260 的网络化计算环境 200 的框图。节点 210 和 260 各自可以是个人计算机（“PC”）、客户端计算机、服务器、主机、膝上型设备、便携式设备、消费电子设备或各种其它类型的计算或处理设备、机器或系统的任一种。以下参照图 11 详细描述了一种计算系统的非限制示例。圆 212、214、262 和 264 表示与其相应节点相关联的 NIC。以下还参照图 11 将一种 NIC 的一个非限制示例描述为网络适配器 1113。

[0032] 如本文所使用的，术语节点是指在网络中（例如网络 202）可唯一寻址或以其它方式可唯一标识并且可用于与该网络中的其它节点通信的任何计算系统、设备或进程。作为示例而非限制，节点可以是个人计算机、服务器计算机、手持型或膝上型设备、平板设备、多处理器系统、基于微处理器的系统、机顶盒、消费电子设备、网络 PC、小型计算机、大型计算机等。以下参照图 11 阐述计算机系统形式的节点 210 的非限制示例。

[0033] 网络 202 和 282 可以是同一网络，可存在于同一或不同的子网上、可彼此在逻辑上或在物理上耦合或隔离、可使用类似或不同的联网技术等。具体地，网络 202 和 282 可以是路由网络，即，包括转发可路由协议分组的路由器的网络。可路由协议通常被认为是用于将数据从一个网络路由到另一个的通信协议。可路由协议的一示例是 TCP/IP。以可路由方式发送数据分组意味着使用可路由传输协议来格式化和 / 或发送数据分组。本领域技术人员应当熟悉可路由协议和路由网络拓朴、系统和体系结构。

[0034] 在一个示例中，网络 202 和 282 可彼此独立，使得在一个网络中存在问题或故障的情况下不会影响到另一个的操作状态。在另一个示例中，可使用三个或更多网络。在其中期望较大的容错程度的示例中，可以采用大量网络连同节点与这些网络的相关联的连通性 - 包括安装在节点上的类似数目的 NIC。

[0035] 与节点 210 相关联的 NIC 212 被示为具有 172.56.48.37 的示例地址并被耦合到网络 1 202。也与节点 210 相关联的 NIC 214 被示为具有 197.71.48.38 的示例地址并被耦合到网络 2 282。与节点 260 相关联的 NIC 262 被示为具有 172.56.48.38 的示例地址并且也被耦合到网络 1 202。也与节点 260 相关联的 NIC 264 被示为具有 197.71.48.39 的示例地址并被耦合到网络 2 282。实际上，这些地址可以是 IPv4 或 IPv6 地址等，或者通常与所使用的协议相关的任何其它类型的网络地址。

[0036] 每个节点可包括一个或多个 NIC。箭头 201 和 230（在图 11 中示为箭头 1114）表示网络 1 202 上的节点 210 与 260 之间的第一通信路由或通路（“路径 A”）。箭头 281 和 283 表示网络 2 282 上的节点 210 与 260 之间的第二通信路由或通路（“路径 B”）。实际上，在环境 200 中的一个或多个网络上的两个或多个节点之间的可存在一条或多条通路。本文所用的术语“通路”被定义为网络中的节点之间的通信路由或通信链路。这样的路由或链

路可以是动态的，因为节点之间的确切路由可随着时间改变。

[0037] 块 216 和 266 表示设置在节点 210 和 260 的每一个上的应用和网络堆栈，包括容错通信（“FT”）驱动器。块 216 的 FT 驱动器被示为具有示例地址 10.0.0.1，而块 266 的 FT 驱动器被示为具有示例地址 10.0.0.2。这些地址通常被认为是虚拟地址。这些地址可以是 IPv4 或 IPv6 地址等，或者任何其它类型的网络或通信地址。FT 驱动器可具有或不具有如以下各个示例中所示的虚拟地址。

[0038] 容错网络堆栈是包括 FT 驱动器的网络堆栈，诸如以下结合图 3 所述的 NETFT 等。结合网络堆栈操作的诸如 NETFT 的 FT 驱动器通常允许节点经由一个或多个网络上的诸如路径 A 和路径 B 的一个或多个通信路径来彼此通信。如果这些通信路径的任一条发生故障，则节点可继续通信，只要至少一条通路可用。这样的通路故障可由于 NIC 的故障或通路的任何元件的故障而导致，包括连接、敷设缆线或其它通信介质（包括射频（“RF”）或红外线（“IR”）等）、路由器、集线器、交换机、防火墙、因特网服务提供商（“IPS”）、任何节点的电源故障、网络的设备或系统等。

[0039] 在一个示例中，通信故障可导致即插即用（“PnP”）事件。PnP 事件指示 NIC 从其节点的移除或介质感测变化。例如，介质感测断开通常由于故障而导致，这使得 NIC 丢失在诸如网络缆线、RF 或 IR 链路等的网络介质上的信号或载波。介质感测断开可由于网络缆线或载波从 NIC 断开或电缆另一端（例如集线器或交换机）的断电而导致。介质感测连接通常是相反的，诸如重新连接缆线、重新对集线器或交换机通电等。也称为连通性事件的此类事件一般是本地事件，因为它们在其自身的节点上或节点附近发生。这些本地连通性事件通常在节点上导致诸如 PnP 事件等的事件指示。

[0040] 在另一示例中，通信故障可通过使用在节点之间发送的心跳 (heartbeat) 分组来检测。这种心跳分组的失败可指示节点之间的通路故障。心跳分组往往被作标记，使得 FT 驱动器一旦接收到它们就可检测它们并对沿网络堆栈向上传递的分组流将其移除。在一个示例中，心跳分组可通过使用路由控制协议（“RCP”）形成 RCP 分组来实现。这些心跳分组可用于验证通路的端对端操作状态。即，通过在路径 A 上将心跳分组从节点 210 发送到节点 260 以及通过节点 210 从节点 260 接收对所发送心跳分组的回复，一般认为路径 A 是端对端可用的。如果心跳失败（未接收到响应于心跳发送的心跳回复），这样的失败可指示路径 A 可能由于网络 1 202 中诸如路由器、交换机、连接等的某些元件的故障、或者由于目的节点自身的故障而不可用。具体地，节点 210 可具有可用 NIC 212 和有效介质感测，这指示其被正确连接到网络，但仍可能由于沿线的某些网络或系统故障而检测到心跳失败。

[0041] 图 3 是示出了示例容错通信驱动器 NETFT 300 的框图。NETFT 300 可被实现为与 NDIS 网络堆栈一起使用以及用于在容忍通路故障的节点之间提供网络通信的 NDIS 微端口驱动器（图 1 中的 160）。即，在每个节点都使用 NETFT 时，尽管通路中的任何组件发生故障，只要至少一条通路仍然可用，两个或更多节点之间的通信就可继续。

[0042] 在一个示例中，将 FT 驱动器实现为 NDIS 微端口驱动器提供了至少两个益处。首先，由于这种 FT 驱动器通常在堆栈中位于任何协议驱动器之下，所以协议可靠性往往要由任何上层可靠协议驱动器来提供，该上层可靠协议驱动器通常不会受到 FT 驱动器所提供的链路级容错的增加的影响。例如，在结合诸如 TCP/IP 驱动器的协议驱动器使用 FT 驱动器时，该 FT 驱动器通常检测到故障的通路，并且在端对端可用通路上与任何协议驱动器无

关地路由数据分组。如果由于切换通路而发生任何分组丢失，则通常在堆栈中位于FT驱动器上的TCP/IP协议驱动器往往检测到这种丢失并执行任意重试或重发操作以确保在分组传送中实现可靠协议。

[0043] 在堆栈中将FT驱动器放置在协议驱动器之下的第二个益处是不会引入协议可路由性的降级。当被如此配置时，FT驱动器对数据分组所执行的任何隧穿(tunneling)操作可采用诸如TCP或UDP的可路由协议，由此除正链路级容错之外确保这种数据可被路由。“可路由地使数据分组隧穿(tunnel)”是使用可路由协议来使传送数据分组隧穿。

[0044] 作为网络堆栈的一部分的NETFT通常经由NDIS或其它网络堆栈接口耦合到软件应用程序。这种耦合通常使得应用程序能够在耦合到堆栈底部的网络上发送和接收分组。在一个示例中，应用程序倾向于对其数据分组使用虚拟地址作为源地址，这种虚拟地址对于NETFT是已知的，并且如以下所述地被映射和传送到网络上的 其它节点。如图3中所示，NETFT包括微端口适配器302(也称为处理元件)、路由数据库304以及一个或多个路由监视器适配器306和隧穿适配器308。

[0045] 隧穿适配器308通常表示本地节点上的一个NIC(或，在某些实例中为虚拟NIC)并且维护用于将分组隧穿到目的节点上的NETFT的插座。通常存在与本地节点上的每个NIC相关联的一个隧穿适配器308，并且每个NIC被耦合到提供了通往另一节点的通路的网络。每个网络可与任何其它网络隔离或不隔离。隧穿适配器308通常源图隧穿协议驱动器相关联并经由NDIS接口通过隧穿协议将数据分组隧穿到其相关联的NIC或从该NIC隧穿。隧穿协议的一个示例是UDP。或者，诸如TCP、IPX或SPX的其它协议可用于进行隧穿。如果相关联的NIC或媒体连接变成不活动，则隧穿适配器308可变成不活动。

[0046] 如NETFT中实现的路由数据库304通常是简单的数据结构，该数据库可位于系统存储器中并包括将一条或多条通路的虚拟地址映射到另一节点上的类似NETFT的条目。在一个示例中，映射是由诸如路由监视器适配器306的路由监视器适配器来表示的，该适配器通常与诸如隧穿适配器308的隧穿适配器相关联。一般诸如路由数据库304的路由数据库将包括对应于每个隧穿适配器的一组路由适配器，每个路由适配器与在关联于隧穿适配器的通路上可到达的不同目的节点相关联。例如，当使用TCP/IP时，数据库可将目的虚拟地址映射到特定远程节点的物理地址。

[0047] 路由数据库304还可包括每条通路的优先级信息。这些优先级信息可用于指示通往另一节点的优选或主要通路和/或包括与通路速率或其它特性相关的信息。优选通路是由NETFT算出的在可能的情况下基于优先级信息和/或通路状态优先其它可能的通路来使用的通路。优先级信息可另外指示用于使通往目标节点的多个通路的使用在通路之间达到通信量负载均衡的一系列负载均衡算法，或者启用某些其它通路优先化方案。

[0048] 表1中示出了一示例路由表数据库304的映射表。

[0049]

目的地址类型	地址	优先级
虚拟	10.0.0.2	--
物理:路径A	172.56.48.38	1
物理:路径B	197.71.48.39	2

[0050] 表1

[0051] 参看表1和图2，表1示出了可由运行在节点216上的NETFT使用的示例映射表。

表 1 示出了虚拟目的地址 10.0.0.2，该虚拟地址被示为对应于节点 266，并被映射到与通往节点 266 的路径 A 相关联的物理地址 172.56.48.38 以及与通往节点 266 的路径 B 相关联的物理地址 197.71.48.39。路径 A 被示为具有第一优先级而路径 B 被示为具有第二优先级。表 1 作为示例而非限制而提供。

[0052] 当从节点 216 向节点 266 发送数据时，这种映射表通常用于通过经由诸如 UDP 的隧穿协议转发分组来将去往虚拟目的地址 10.0.0.2 的分组隧穿到物理地址 172.56.48.38，由此在路径 A 上将分组从节点 216 隧穿到节点 266。可在路由数据库（图 3 的 304）中为两个节点之间建立的每组通路创建一个这样的映射表。这种映射表可以通过各种形式实现，使用各种优先级方案和 / 或存储包括通路操作状态的其它信息。表 1 中所示的映射表结构、通路数目、地址格式等作为示例而非限制而提供。

[0053] 本地节点虚拟地址、远程节点虚拟地址以及优先级和其它通路信息通常通过带外 (out-of-band) 机制来向节点提供，并经由其 NDIS 接口传递给 NETFT。这种带外机制可与使用管理应用来指定信息的系统管理器一样简单，或者其可以是自动化系统等。这种带外机制对于本领域技术人员是公知的。

[0054] 如图 3 中所示，微端口适配器 302（也称为驱动器的处理元件）通常解析沿网络堆栈向下流动的数据分组、检查该分组的目的虚拟地址以及使用来自路由数据库 304 的信息来确定将该数据分组隧穿通过哪个隧穿适配器 308。传入分组或沿堆栈向上流动的数据分组朝着目的虚拟地址沿堆栈向上转发，并且隧穿协议已在先前移除了隧穿分组的报头。具体地，隧穿适配器 308 检查传入分组并将心跳分组转发到路由监视器适配器 306，以及通过微端口适配器 302 沿堆栈向上转发其它分组。使用隧穿协议来使分组隧穿以及如何通过协议驱动器添加和移除协议报头的方面对于本领域技术人员而言是公知的。

[0055] 路由监视器适配器 306 通常表示在由相关联的隧穿适配器标识的特定通道上可访问的远程节点。路由监视器适配器 306 通常将提供远程节点的物理地址，该物理地址 306 还对应于通往远程节点的特定通路。这种物理地址通常被用于路由数据库 304 中的映射。对于通往远程节点的每个不同通路，通常存在一个远程监视器适配器，每个路由监视器适配器与表示通路的隧穿适配器相关联。在一个示例中，再次参看图 2，节点 210 被示为通过两条通路耦合到节点 260，其一通过网络 1 202（“路径 A“），而另一条通过网络 2 282（“路径 B”）。运行在节点 210 上的 NETFT 可包括第一路由监视器适配器（“RMA-A”），该适配器提供了远程节点 260 的与其 NIC 262 相关联的物理地址 172.56.48.38。RMA-A 可与节点 210 上的第一隧穿适配器（“TA-A”）相关联，其中该节点可与路径 A 相关联。节点 210 上的 NETFT 还可包括第二路由监视器适配器（“RMA-B”），该适配器提供了远程节点 260 的与其 NIC 264 相关联的物理地址 197.71.48.39。RMA-B 可与节点 210 上的第二隧穿适配器（“TA-B”）相关联，其中该节点可与路径 B 相关联。

[0056] 参看图 3，路由监视器适配器 306 通常监视通往远程节点的通路的健康状况，并在路由数据库 304 中指示有故障的或不可用的通路。监视通常包括接收任何事件指示和 / 或通知任何心跳失败以及相应地更新数据库 304。在一个示例中，指示 NIC 或介质连接的故障的事件可导致隧穿适配器 308 的禁用。在另一示例中，心跳失败可导致与心跳失败的特定远程节点相关的路由监视器适配器 306 的禁用。

[0057] 图 4 是示出了包括 NETFT 300 和应用 402 的示例容错通信体系结构 216 的框图。

在此示例中，应用 402 使用虚拟源地址 217 和表示目的节点的虚拟目的地址来经由堆栈将数据分组发送到NETFT 300。这种传出数据分组经由路径480从应用通过网络堆栈流动到驱动器 300。驱动器 300 通常使用存储在路由数据库中的优先级信息和通路操作状态信息来确定每个分组应当采用哪条可能的通路，并且在所选的通路上使用适当的物理源地址 422 或 424 来将该分组隧穿到目标节点。

[0058] 应用 402 可通过 NETFT 经由 TCP 协议来发送数据分组，如图 4 中所示。或者可使用 UDP 或任何其它协议。而且，如所示，NETFT 300 可使用 UDP 协议来将分组隧穿到目标节点。或者，TCP 或任何其它协议可用于进行隧穿。此外，替代示例可不使用微端口适配器或 NDIS 驱动器但可使用其它机构或体系结构来执行类似功能。最后，网络堆栈的各个元件等可以在用户模式或内核模式中运行，或者如所示或其它方式，在具有或不具有等效操作模式的系统上运行。

[0059] 图 5 是示出了数据流过包括经由网络 1 202 上的路径 A 和网络 2 282 上的路径 B 耦合的源节点 216 和目的节点 266 的容错通信环境 500 的流程图。在此示例环境 500 中，数据被示为从在节点 216 上运行的应用发送到在节点 266 上监听目的虚拟地址的应用。数据分组使用 TCP 协议沿节点 216 上运行的网络堆栈向下流到 NETFT 中，如路径 501 所示。如所示，假设路径 A 是所选通路，NETFT 将数据分组从由应用正在使用的源虚拟地址映射到路径 A，并通过 UDP 使用目的节点 266 的路径 A 的物理目的地址来如路径 501 进一步所示地将该数据隧穿出节点 216 的 NIC 1，然后经由链路 201 进入网络 1 202。该数据随后流过网络 1 202、通过链路 203、接着到达节点 266、如路径 503 所示地沿节点 266 上运行的网络堆栈向上流动。该数据随后流过 UDP 协议（与发送侧用作隧穿协议的协议一样）驱动器，其中从数据分组剥离 UDP 协议报头，随后将其传递到运行在节点 266 上的 NETFT。NETFT 随后将这些数据分组沿堆栈向上转发到正监听目的虚拟地址的应用。响应往往以相反次序流动。

[0060] 图 6 是示出了在外加若干可能的通信故障 610、612、620 和 630 的情况下数据流过图 5 中所示的容错通信环境 500 的流程图。其它通信故障也是可能的。故障 610 指示在发送节点 216 上运行的 NIC 1 的故障。这种故障可能在 NIC 1 被从节点移除、NIC 1 的驱动器发生故障、NIC 1 自身发生故障等情况下发生。故障可由 NETFT 经由诸如 PnP 事件等的事件指示和 / 或心跳失败来检测。在这种情形中，路径 A 通常被认为已发生故障，并且 NETFT 将选择替代的端对端可用通路。端对端可用通路通常可始终成功地将数据从源节点和应用递送到目的节点和应用。

[0061] 故障 620 指示与节点 216 耦合的网络介质的故障。此故障可能由于缆线从 NIC1 断开、由于缆线从网络 1 202 的某一设备断开、由于在网络一侧上缆线所连接的设备断电或发生故障等而导致。这种类型的故障还可由 NETFT 经由诸如 PnP 事件等的事件指示和 / 或心跳失败以及所选的替代通路来检测。

[0062] 故障 630 指示在网络 202 内导致数据分组无法到达目的节点 266 的某一类故障。在此故障的情形中，发送节点 216 仍耦合到网络 202 并存在适当的介质感测指示，但是路径 A 已进一步使网络中断。在这种故障下，如果本地指示显示通往网络 202 的连通性良好，则运行在发送节点 216 上的 NETFT 可能无法通过事件指示检测到该故障，但是可通过路径 A 的心跳失败检测到该故障。

[0063] 链路 203 的故障 622 以及在接收节点 266 上运行的 NIC 1 的故障 612 往往类似于

针对节点 216 所示的相应故障。虽然并不在节点 216 本地的这些故障可能无法通过事件指示来检测，但是可通过心跳失败来检测。

[0064] 这些故障的任一个以及其它故障可通过在节点 216 上运行的 NETFT 来检测，并可导致其选择替代的端对端可用通路，诸如在网络 2 282 上的路径 B。在此示例中，如图 6 中所示，NETFT 将数据沿替代路径 681 向下隧穿并通过网络 2 282 到达接收节点 266。如果路径 A 上的故障情况被修复且端对端操作状态恢复，则在发送节点 216 上运行的 NETFT 可检测到该恢复并再次使用路径 A。此外，从节点 266 返回到节点 216 的任何响应可以类似的容错方式通过 NETFT 来隧穿。

[0065] 图 7 是示出了容错通信驱动器 NETFT 700 的另一示例的框图。此示例类似于 图 3 中所示示例，但是包括以下所述的变化。在此示例中，软件应用程序可能无需使用虚拟地址。相反，应用程序可使用物理地址来使数据分组寻址到目标节点。

[0066] 协议适配器 710 通常耦合到微端口适配器 702（也称为驱动器的处理元件）和 NIC 微端口适配器（未示出）。通常，对安装节点上的每一个 NIC 存在一个协议适配器，每个协议适配器经由其 NIC 适配器与 NIC 相关联。由于每个协议适配器与一 NIC 相关联，所以它也与耦合到该 NIC 的通路相关联。协议适配器 710 可用于经由处理元件 702 从应用接受数据分组，并将该数据分组传递到相关联的 NIC 而无需进行隧穿。

[0067] 处理元件 702 通常解析沿网络堆栈向下流动的数据分组、检查这些分组的物理目的地址并使用来自路由数据库 704 的信息来确定该分组是可在协议适配器 710 上转发还是需要通过隧穿适配器 308 隧穿到目标节点。一般而言，如果由物理目的地址指示的通路是端对端可用的，则数据分组将通过该通路发送。可选择将分组在其上隧穿的其它或替代的通路。

[0068] 在此示例中，路由数据库 704 维护物理目的地址和通路的映射以及如上所述的优先级和其它信息。表 2 中示出了示例路由数据库 704 的映射表。

[0069]

目的地址类型	地址	优先级
物理 :路径 A	172.56.48.38	1
物理 :路径 B	197.71.48.39	2

[0070] 表 2

[0071] 参看表 2 和图 2，表 2 示出了可由在节点 216 上运行的 NETFT 所用的示例映射表。表 2 示出了包括与通往节点 266 的路径 A 相关联的物理目的地址 172.56.48.38 和与通往节点 266 的路径 B 相关联的物理目的地址 197.71.48.39 的映射。路径 A 被示为具有第一优先级，而路径 B 具有第二优先级。

[0072] 当从节点 216 向节点 266 发送数据时，这种映射表通常被用于转发（或者按需进行隧穿）正被发往节点 266 的物理目的地址 172.56.48.38 的数据分组。如果与初始目的地址相关联的通路是可用的，则数据分组往往被转发到该目的地址而不进行隧穿。如果该通路不可用，则该数据分组通过替代的通路经由隧穿被发送到节点 266 的物理目的地址 197.71.48.39。NETFT 700 的其它方面一般类似于针对图 3 所述的 NETFT。

[0073] 图 8 是示出了包括 NETFT 700 和应用 402 的示例容错通信体系结构 216 的框 图。在此示例中，应用 402 使用物理源地址和表示目的节点的物理目的地址 801 经由堆栈来将数据分组发送到NETFT 700。这种传出数据分组经由路径880从应用通过网络堆栈流动到驱

动器 700。该驱动器 700 通常使用存储在路由数据库中的优先级信息和通路操作状态信息确定每个分组应当采用哪条可能的通路，并且通过由初始物理目的地址指示的路径将该分组转发到目标节点，或者如果该通路并非端对端可用，则通过如在本示例中由路由 882 和 NIC2 892 所指示的替代通路来使分组隧穿。

[0074] 应用 402 可经由 TCP 协议通过 NETFT 700 发送数据分组，如图 8 所示。可使用 UDP 或任何其它协议作为替代。而且，如所示，NETFT 700 可使用 UDP 协议来将分组隧穿到目标节点。或者，TCP 或任何其它协议可用于进行隧穿。此外，其它示例可不利用 NDIS 驱动器，但可使用其它机构或体系结构来执行类似功能。最后，网络堆栈等的各个要素可以在用户模式或内核模式中运行，或者如所示或以其它方式，在具有或不具有等效操作模式的系统上运行。

[0075] 图 9 是示出了数据流过包括经由网络 1 202 上的路径 A 和网络 2 282 上的路径 B 耦合的源节点 816 和目的节点 966 的容错通信环境的流程图。在此示例环境 900 中，数据被示为从在节点 216 上运行的应用发送到在节点 266 上监听目的物理地址的应用。数据分组使用 TCP 协议沿在节点 216 上运行的网络堆栈向下流到 NETFT 中，如路径 901 所示。如所示，假设路径 A 是所选通路，NETFT 使用由应用所提供的物理目的地址来将数据分组经由节点 216 的 NIC 1 在路径 A 上转发以及经由链路 201 在网络 1 202 转发。数据随后流过网络 1 202、通过链路 203 然后到达节点 966、如由路径 903 所示地沿在节点 966 上运行的网络堆栈向上流动。数据随后流过 NETFT 和协议驱动器（用于与发送侧用作发送协议相同的协议的协议驱动器）并到达应用。响应往往以反向次序流动。

[0076] 图 10 是示出了在外加了若干可能的通信故障 1010、1012、1020、1022 和 1030 的情况下数据流过图 9 中所示的容错通信环境 900 的流程图。其它通信故障也是可能的。故障 1010 指示在发送节点 816 上运行的 NIC 1 的故障。这种故障可能在 NIC1 被从节点移除、其 NIC 驱动器发生故障、NIC 自身发生故障等的情况下发生。该故障可由 NETFT 经由诸如 PnP 事件等的事件指示和 / 或心跳失败来检测。在这种情形中，路径 A 通常被认为已发生故障，并且 NETFT 将选择替代的端对端可用通路。

[0077] 故障 1020 指示与节点 816 的 NIC 1 耦合的网络介质的故障。此故障可由于缆线从 NIC 1 断开、由于缆线从网络 1 202 的某一设备断开、由于在网络一侧上缆线 所连接的设备断电或发生故障等而导致。此类故障还可由 NETFT 经由诸如 PnP 事件等的事件指示和 / 或心跳失败及所选的替代通路来检测。

[0078] 故障 1030 指示在网络 202 内导致数据分组无法到达目的节点 966 的某一类故障。在此故障的情形中，发送节点 816 仍可耦合到网络 202 并存在适当的介质感测指示，但是路径 A 已进一步使网络中断。在这种故障下，如果本地指示显示通往网络 202 的连通性良好，则运行在发送节点 816 上的 NETFT 可能无法通过诸如 PnP 事件等的事件指示检测到该故障，但是可通过路径 A 的心跳失败检测到该故障。

[0079] 链路 203 的故障 1022 以及在接收节点 966 上运行的 NIC 1 的故障 1012 往往类似于针对节点 816 所示的相应故障。虽然并非节点 816 本地的这些故障可能无法通过事件指示来检测，但是可通过心跳失败来检测。

[0080] 这些故障的任一个以及其它故障可由在节点 816 上运行的 NETFT 来检测，并可导致其选择替代的端对端可用通路，诸如在网络 2 282 上的路径 B。在此示例中，如图 10 中所

示，NETFT 将数据沿替代路径 1081 向下隧穿并通过网络 2 282 到达接收节点 966。如果路径 A 上的故障情况被修复且端对端操作状态被恢复，则在发送节点 816 上运行的 NETFT 可检测到该恢复并再次使用路径 A。此外，从节点 966 返回到节点 816 的任何响应可取决于路径 A 和路径 B 的操作状态以类似的容错方式由其 NETFT 来隧穿。

[0081] 图 11 是示出了其中可实现上述技术的示例性计算环境 1100 的框图。合适的计算环境可使用各种通用或专用系统来实现。公知系统的示例可包括但不限于：个人计算机（“PC“）、手持型或膝上型设备、基于微处理器的系统、微处理器系统、服务器、工作站、消费类电子设备、机顶盒等。

[0082] 计算环境 1100 通常包括耦合到各种外围设备 1102、1103、1104 等的计算设备 1101 形式的通用计算系统。系统 110 可经由一个或多个 I/O 接口 1112 耦合到各种输入设备 1103，包括键盘和诸如鼠标或跟踪球的定点设备。计算设备 1101 的组件可包括一个或多个处理器（包括中央处理单元（“CPU”）、图形处理单元（“GPU”）微处理器（“μP”）等）1107、系统存储器 1109 和通常耦合各种组件的系统总线 1108。处理器 1107 通常处理或执行各种计算机可执行指令以控制计算设备 1101 的操作并经由诸如网络连接 1114 等的各种通信连接与其它电子和 / 或计算设备、系统或环境（未示出）通信。系统总线 1108 表示任意数量的各类总线结构，包括存储器总线或存储器控制器、外围总线、串行总线、加速图形端口、处理器或使用各种总线体系结构的任一种的局部总线等。

[0083] 系统存储器 1109 可包括诸如随机存取存储器（“RAM”）的易失性存储器和 / 或诸如只读存储器（“ROM”）或闪存（“FLASH”）的非易失性存储器的形式的计算机可读介质。基本输入 / 输出系统（“BIOS”）可以非易失性等形式来存储。系统存储器 1109 通常存储数据、计算机可执行指令和 / 或包括可直接访问或正由一个或多个处理器 1107 操作的计算机可执行指令的程序模块。

[0084] 大容量存储设备 1104 和 1110 可通过耦合到系统总线而耦合到计算设备 1101 或结合到计算设备 1101。这种大容量存储设备 1104 和 1110 可包括对可移动的非易失性磁盘（例如“软盘”）1105 读和 / 或写的磁盘驱动器，和 / 或对诸如 CD ROM、DVD ROM 1106 的非易失性光盘读和 / 或写的光盘驱动器。或者，诸如硬盘 1110 的大容量存储设备可包括不可移动存储介质。其它大容量设备可包括存储卡、存储棒、带式存储器等。

[0085] 大量计算机程序、文件、数据结构等可被存储在硬盘 1110、其它存储设备 1104、1105、1106 和系统存储器 1109（通常受可用空间的限制）上，作为示例，这些大量计算机程序、文件、数据结构等包括：操作系统、应用程序、数据文件、目录结构和计算机可执行指令。

[0086] 诸如显示设备 1102 的输出设备可经由诸如视频适配器 1111 耦合到计算设备 1101。其它类型的输出设备可包括打印机、音频输出、触觉设备或其它感官输出机构等。输出设备可允许计算设备 1101 与操作员或者其它机器或系统交互。用户可经由诸如键盘、鼠标、操纵杆、游戏垫、数据端口等任何数目的不同输入设备 1103 与计算环境 1100 接口。这些及其它输入设备可经由可耦合到系统总线 1108 的输入 / 输出接口 1112 来耦合到处理器 1107，并且可通过诸如并行端口、游戏端口、通用串行总线（“USB”）、火线、红外端口等的其它接口和总线结构来耦合。

[0087] 计算设备 1101 可通过一个或多个局域网（“LAN”）、广域网（“WAN”）、存储域网（“SAN”）、因特网、无线电链路、光学链路等经由到一个或多个远程计算设备的通信连接而

在网络化环境中运行。计算设备 1101 可经由网络适 1113 等、或者经由调制解调器、数字用户线路（“DSL”）链路、综合业务数字网（“ISDN”）链路、因特网链路、无线链路等耦合到网络。

[0088] 诸如网络连接的通信连接 1114 通常提供到诸如网络的通信介质的耦合。通信介质通常使用诸如载波或其它传输机制的调制数据信号提供计算机可读和计算机可执行指令、数据结构、文件、程序模块和其它数据。术语“调制数据信号”通常表示以在该信号中编码信息的方式设置或改变其特征中的一个或多个的信号。作为 示例而非限制，通信介质可包括诸如有线网络或直接线连接等的有线介质，以及诸如声波、射频、红外线和其它无线通信机制的无线介质。

[0089] 本领域技术人员将认识到用于提供计算机可读和计算机可执行指令及数据的存储设备可分布在网络上。例如，远程计算机或存储设备可以采用软件应用程序和数据的形式来存储计算机可读和计算机可执行指令。本地计算机可经由网络访问远程计算机或存储设备，以及下载软件应用程序或数据的部分或全部并可执行任何计算机可执行指令。或者，本地计算机可按需下载软件或数据的片段，或者通过在本地计算机上执行部分指令而在远程计算机和 / 或设备上执行另一部分指令来分布式地处理软件。

[0090] 本领域技术人员还将认识到，通过使用常规技术，软件的计算机可执行指令的全部或部分可通过诸如数据信号处理器（“DSP”）、可编程逻辑阵列（“PLA”）、离散电路等的专用电子电路来实现。术语“电子装置”可包括包含任何软件、固件等的计算设备或消费类电子设备、或者不包含软件、固件等的电子设备或电路。

[0091] 术语“固件”通常是指维护在诸如 ROM 的电子设备中的可执行指令、代码或数据。术语“软件”通常是指维护在任何形式的计算机可读介质中或其上的可执行指令、代码、数据、应用、程序等。术语“计算机可读介质”通常是指系统存储器、存储设备及其相关联的介质、通信介质等。

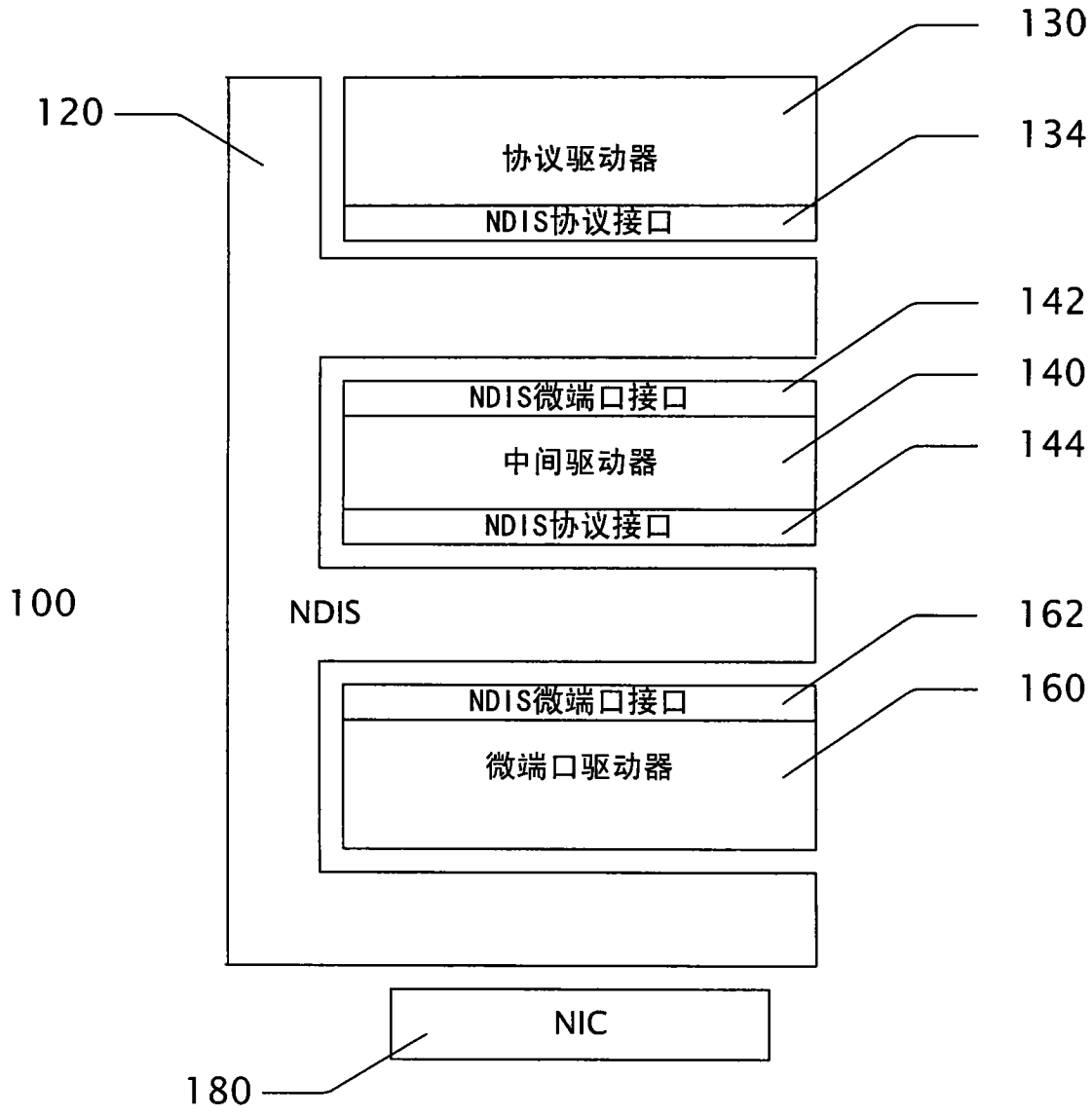

130
120
协议驱动器
134
NDIS协议接口
142
NDIS微端口接口
140
中间驱动器
144
NDIS协议接口
100
NDIS
162
NDIS微端口接口
160
微端口驱动器
NIC
180

图 1

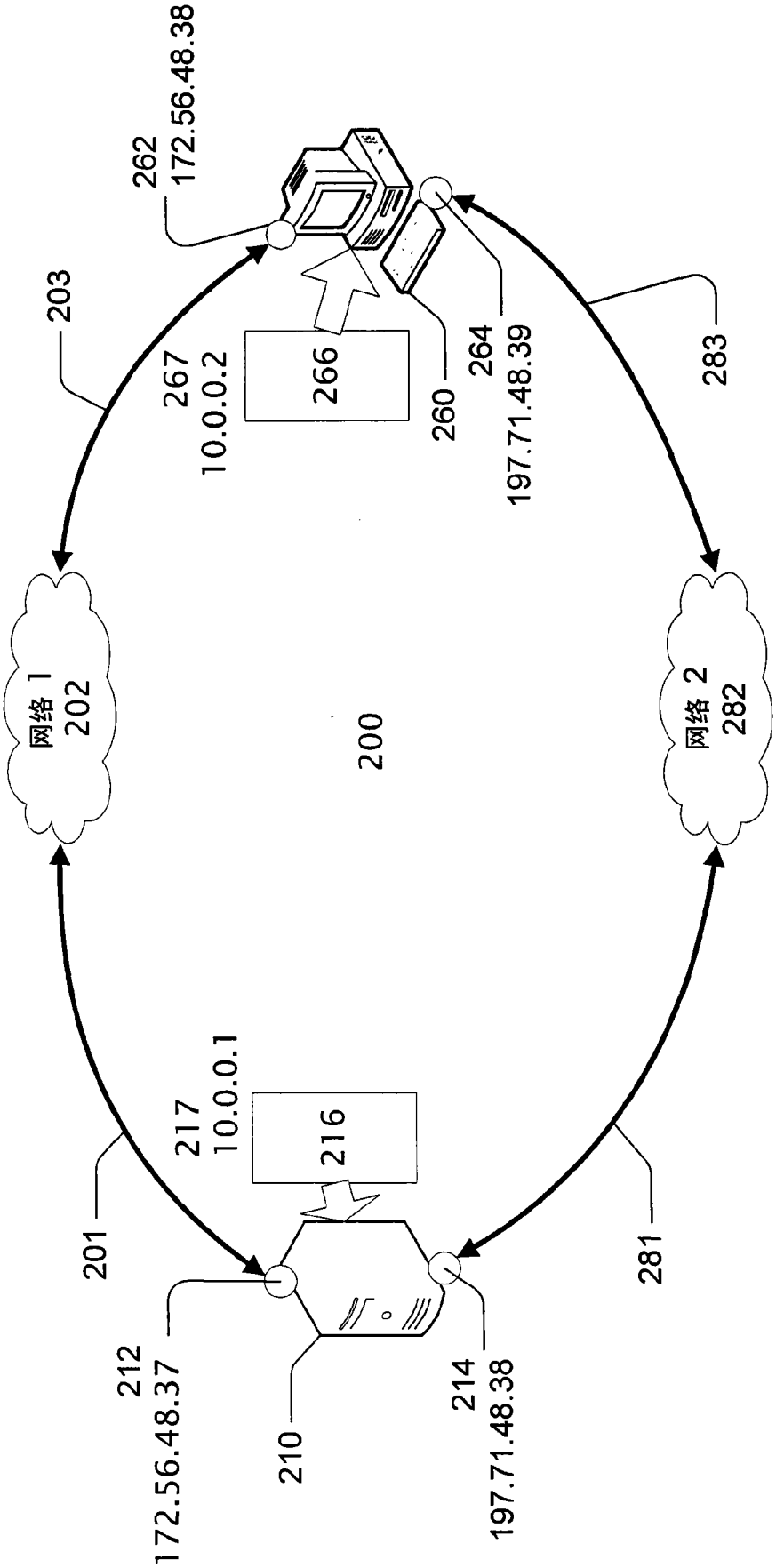

262
172.56.48.38
203
267
10.0.0.2
266
260
264
197.71.48.39
283
网络 1
202
200
网络 2
282
217
10.0.0.1
216
201
281
212
172.56.48.37
210
214
197.71.48.38

图 2

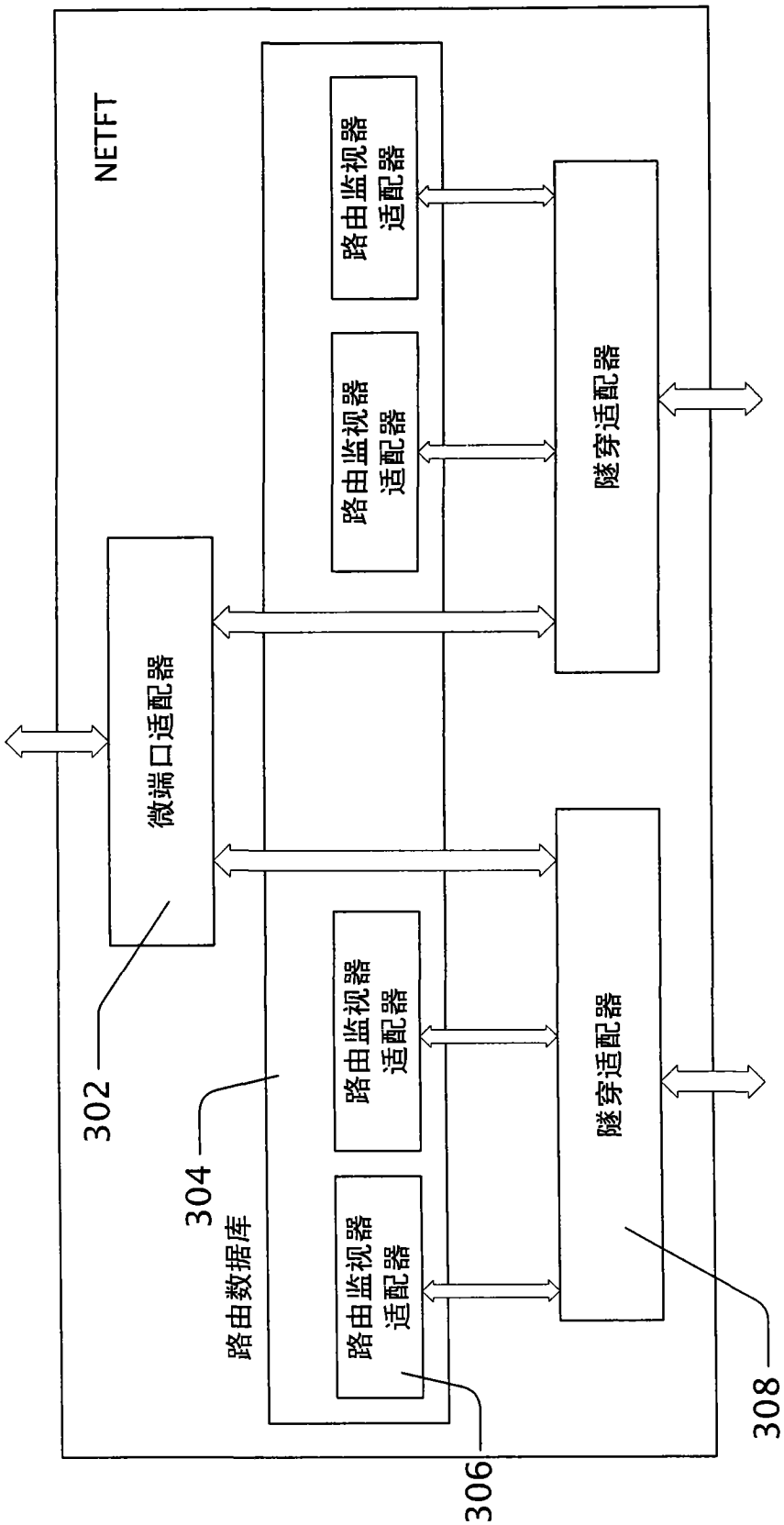

NETFT
路由监视器适配器
路由监视器适配器
隧穿适配器
微端口适配器
302
304
路由数据库
路由监视器适配器
路由监视器适配器
306
隧穿适配器
308
300

图 3

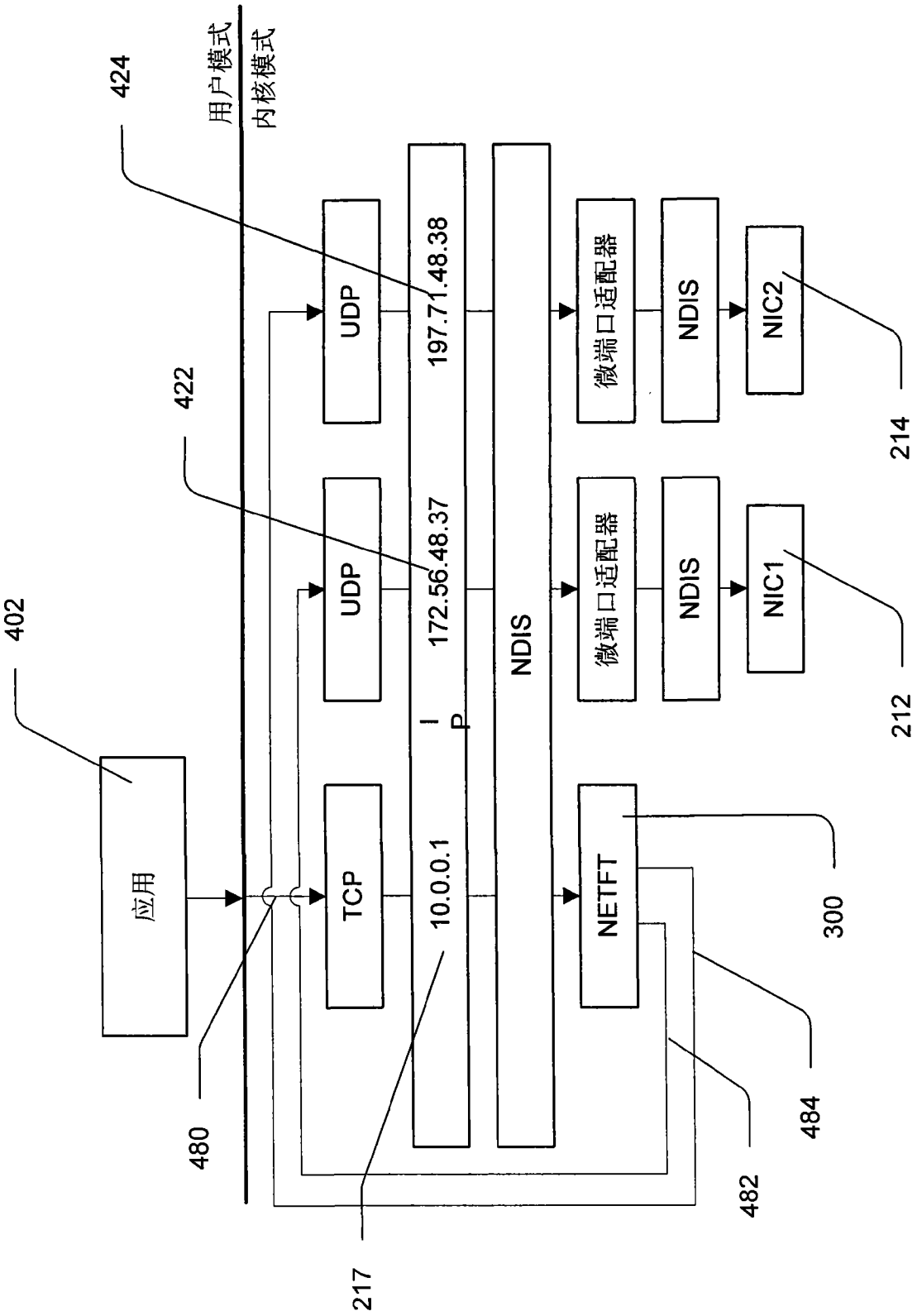

用户模式
内核模式
424
422
402
应用
TCP
UDP
UDP
IP
10.0.0.1
172.56.48.37
197.71.48.38
NDIS
NETFT
微端口适配器
微端口适配器
NDIS
NDIS
NIC1
NIC2
480
482
484
300
212
214
216
217

图 4

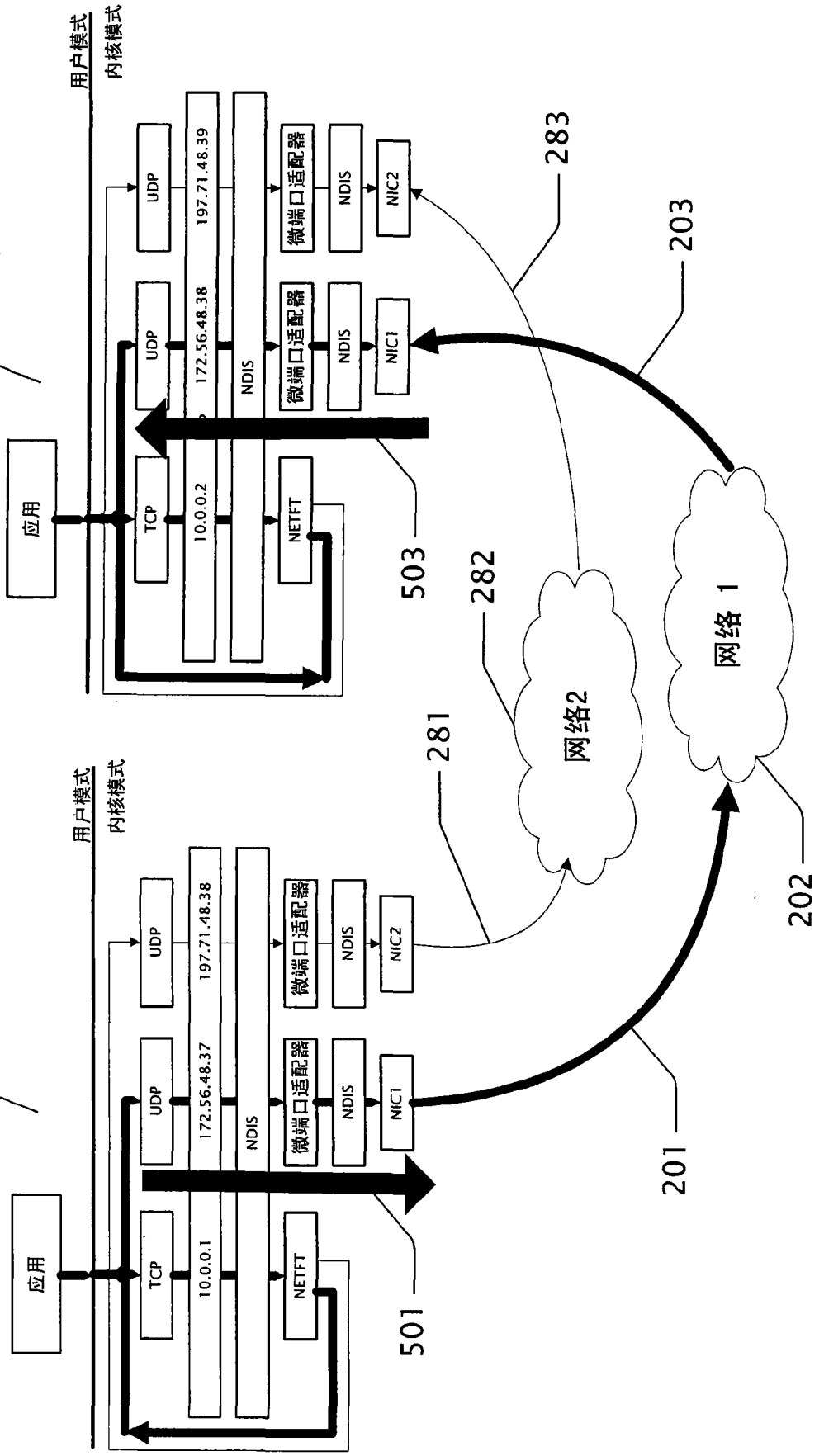

用户模式
内核模式
266
应用
TCP
UDP
10.0.0.2
172.56.48.38
197.71.48.39
NDIS
NETFT
微端口适配器
NDIS
NIC1
NIC2
503
283
203
282
281
网络2
网络 1
202
201
216
应用
TCP
UDP
10.0.0.1
172.56.48.37
197.71.48.38
NDIS
NETFT
微端口适配器
NDIS
NIC1
NIC2
501
500

图 5

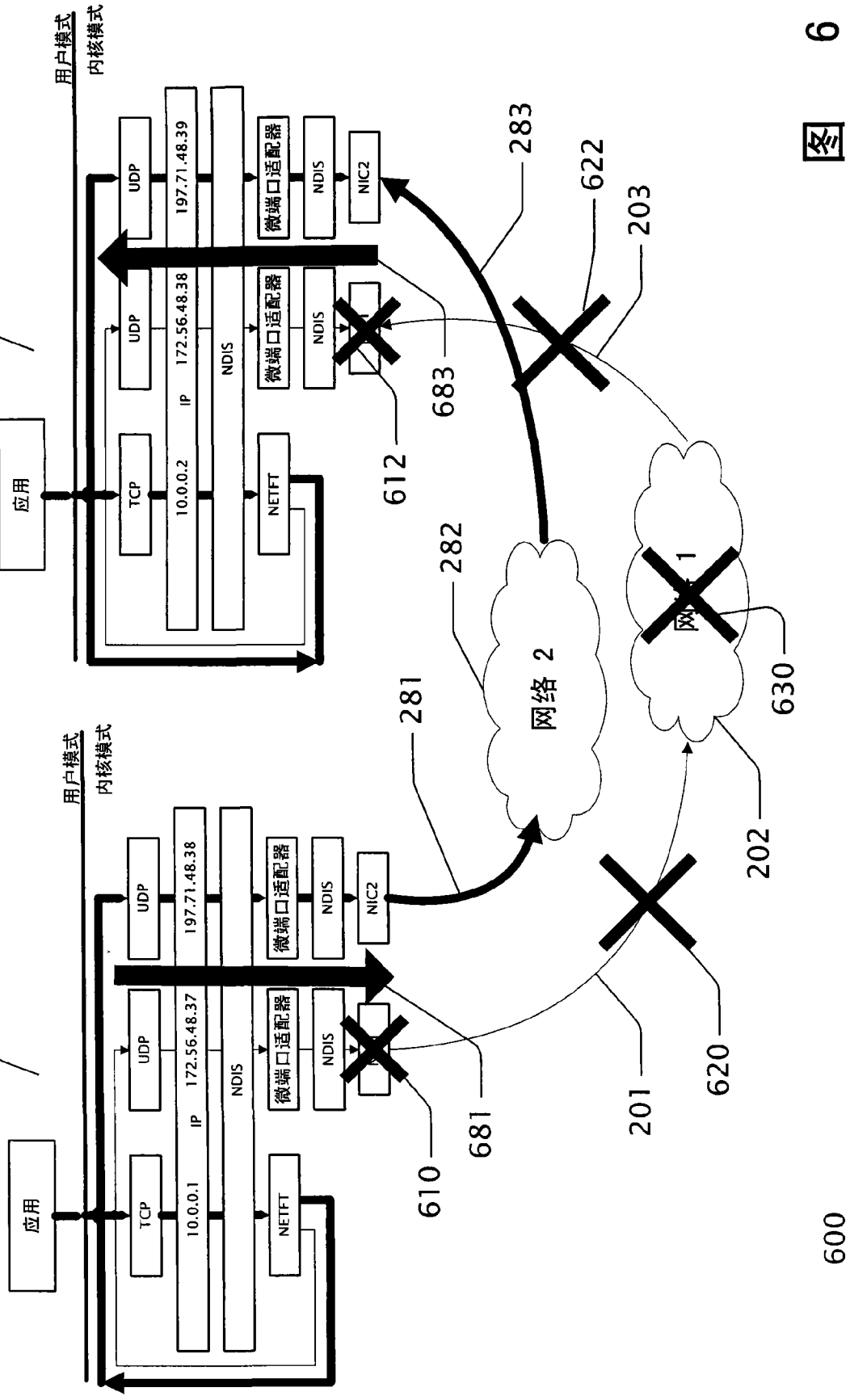

用户模式
内核模式
应用
TCP
UDP
IP
10.0.0.2
172.56.48.38
197.71.48.39
NDIS
微端口适配器
NETFT
NIC2
266
216
610
612
620
622
630
681
683
201
202
203
281
282
283
网络 2
网络 1
600

图 6

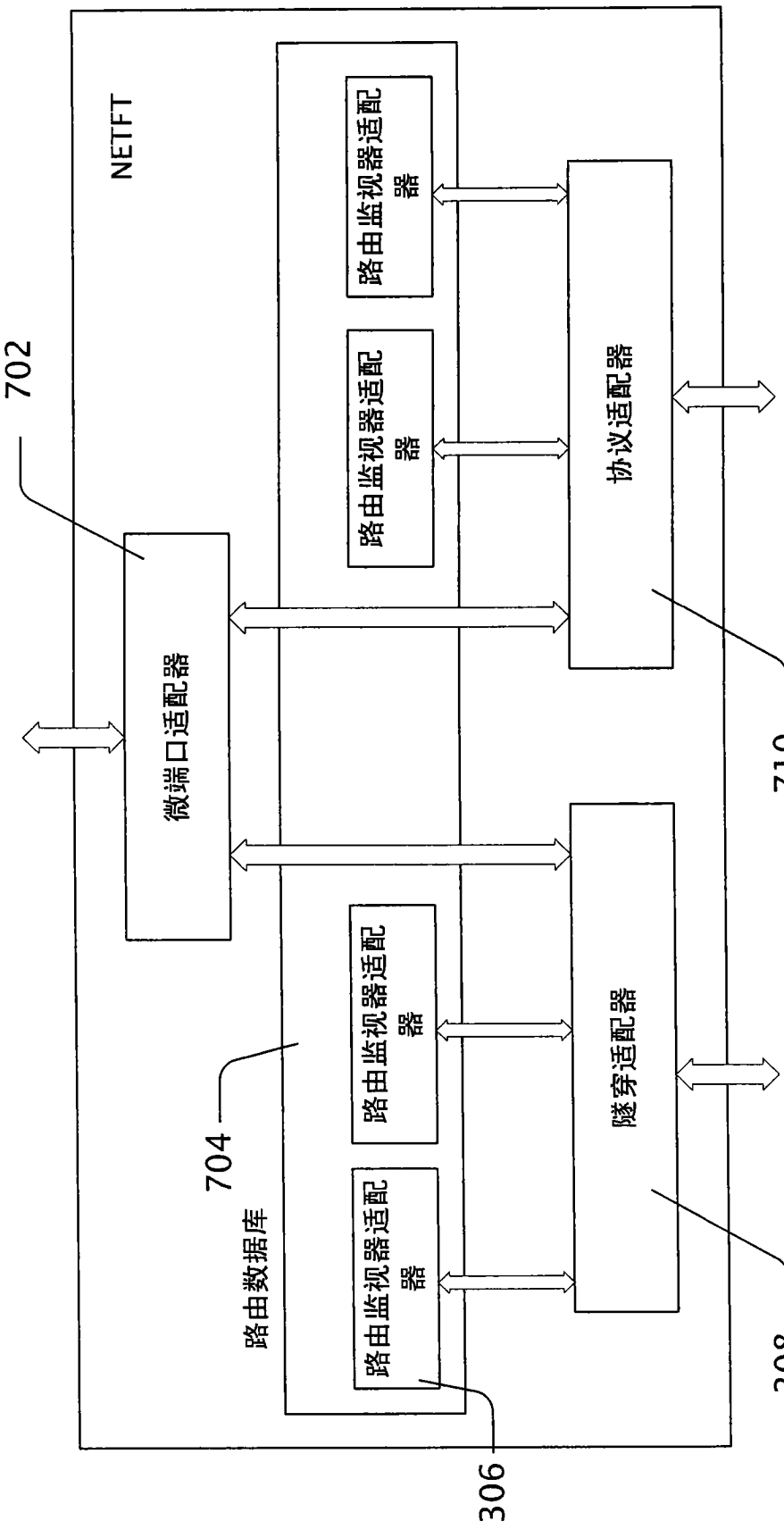

NETFT
路由监视器适配器
路由监视器适配器
协议适配器
微端口适配器
路由监视器适配器
路由监视器适配器
隧穿适配器
路由数据库
702
704
710
700
306
308

图 7

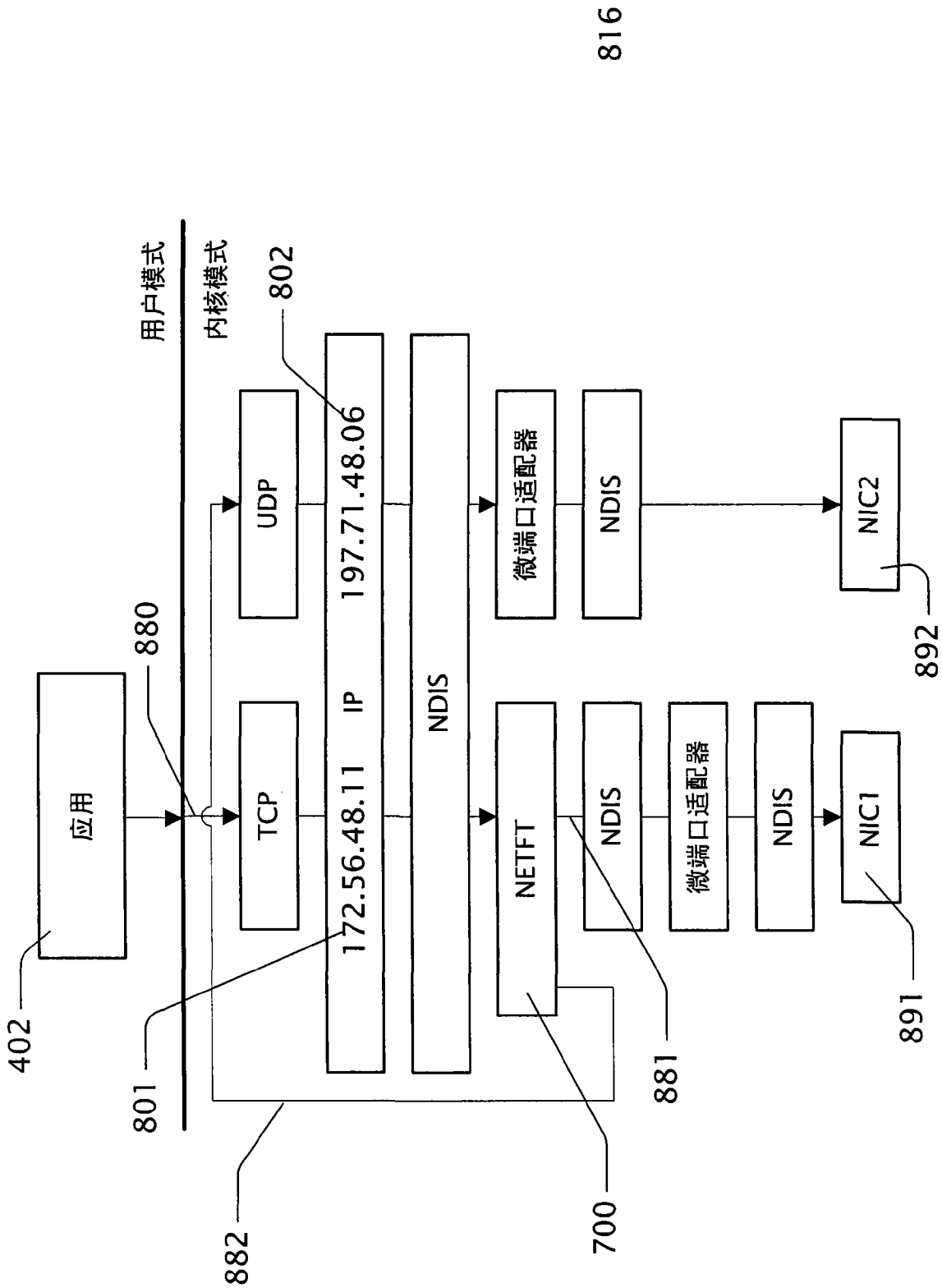

816
用户模式
内核模式
802
880
应用
UDP
TCP
IP
197.71.48.06
172.56.48.11
NDIS
微端口适配器
NDIS
NIC2
892
NETFT
NDIS
微端口适配器
NDIS
NIC1
891
402
801
881
882
700

图 8

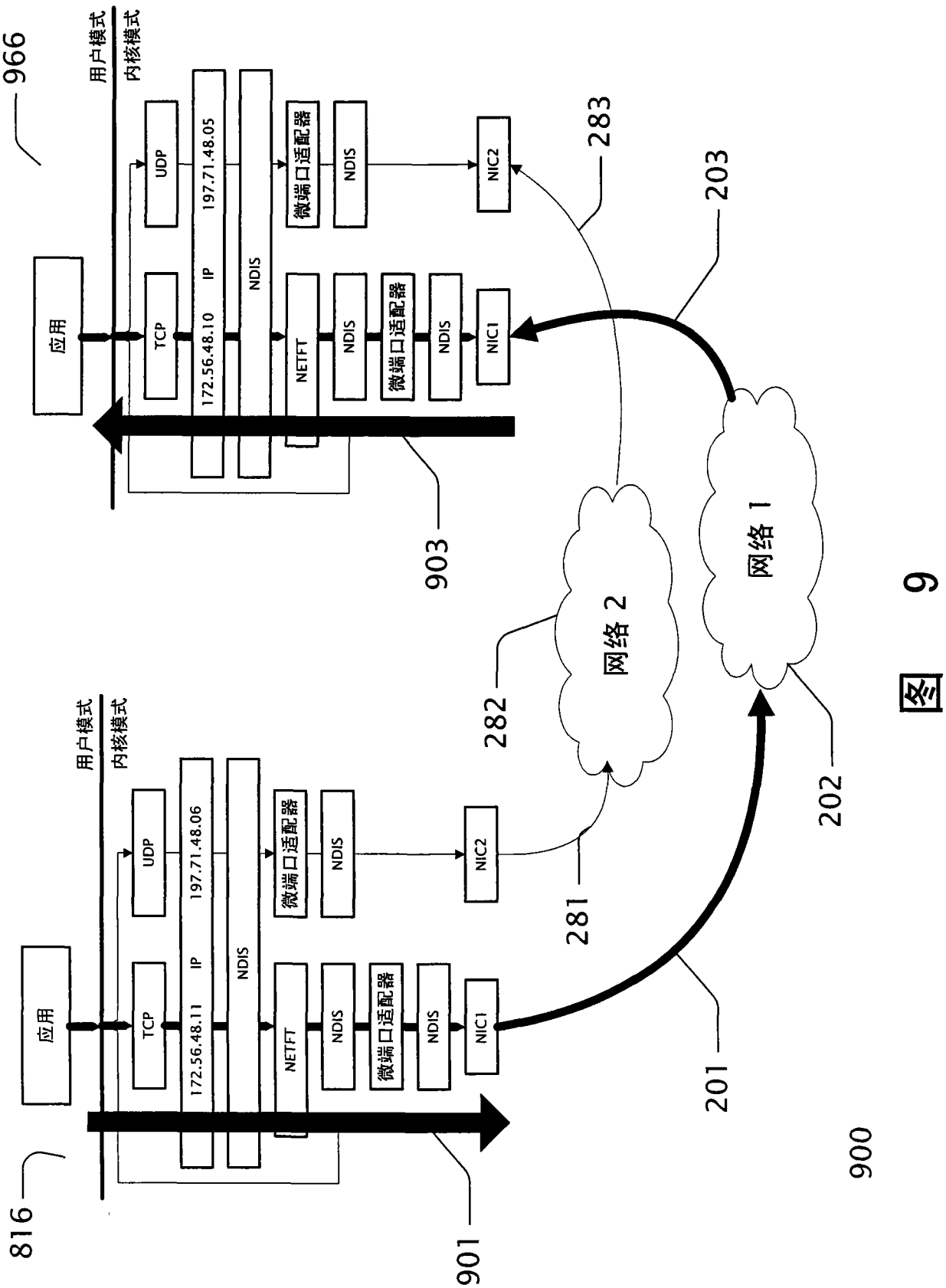

966
用户模式
内核模式
应用
TCP
UDP
IP
172.56.48.10
197.71.48.05
NDIS
NETFT
微端口适配器
NIC1
NIC2
903
283
203
网络 2
网络 1
282
281
202
201
816
172.56.48.11
197.71.48.06
901
900

图 9

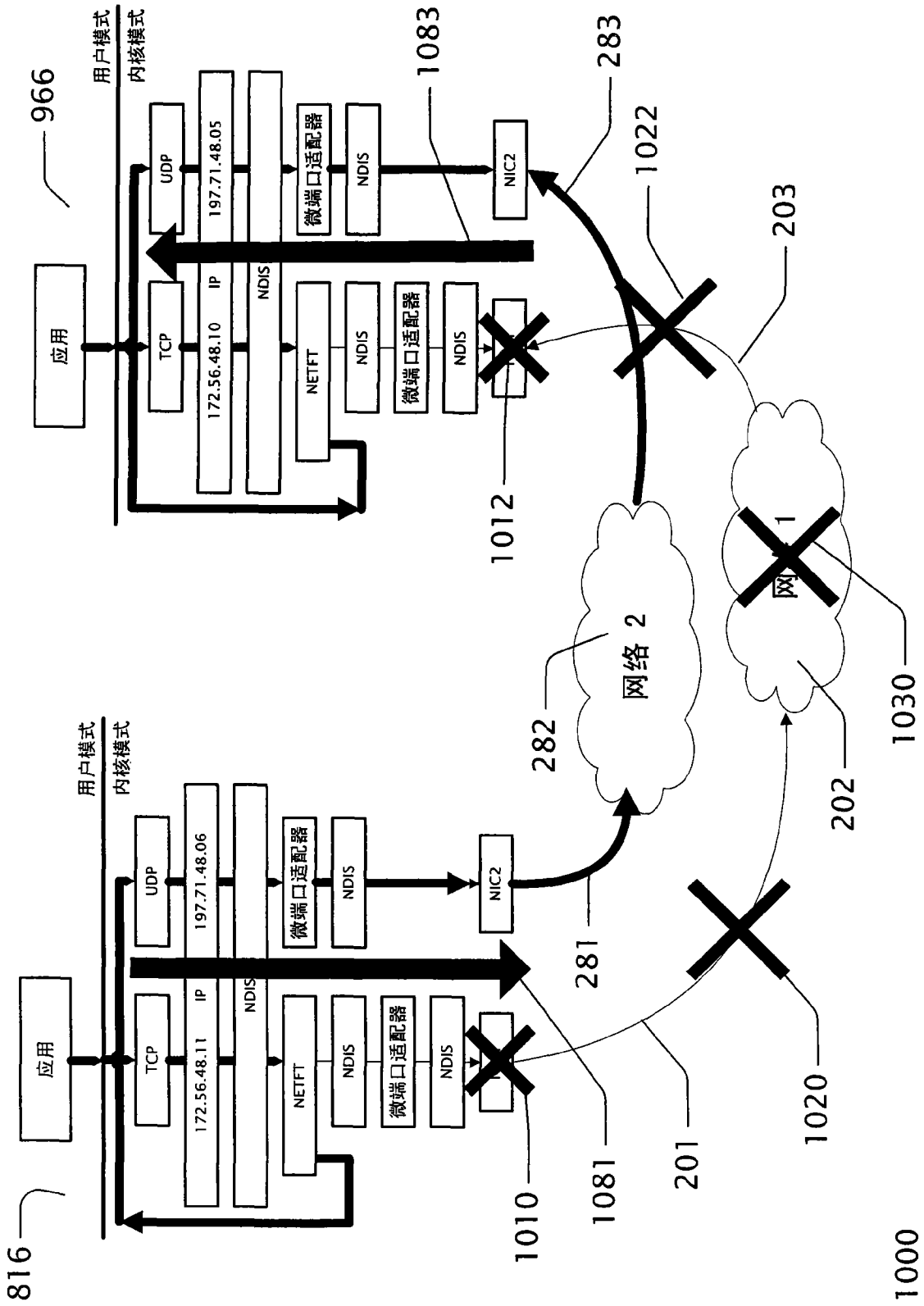

966
用户模式
内核模式
应用
TCP
UDP
IP
172.56.48.10
197.71.48.05
NDIS
NETFT
微端口适配器
NIC2
1083
283
1022
203
1012
网络 2
网络 1
282
1030
202
816
172.56.48.11
197.71.48.06
281
1010
1081
201
1020
1000

图 10

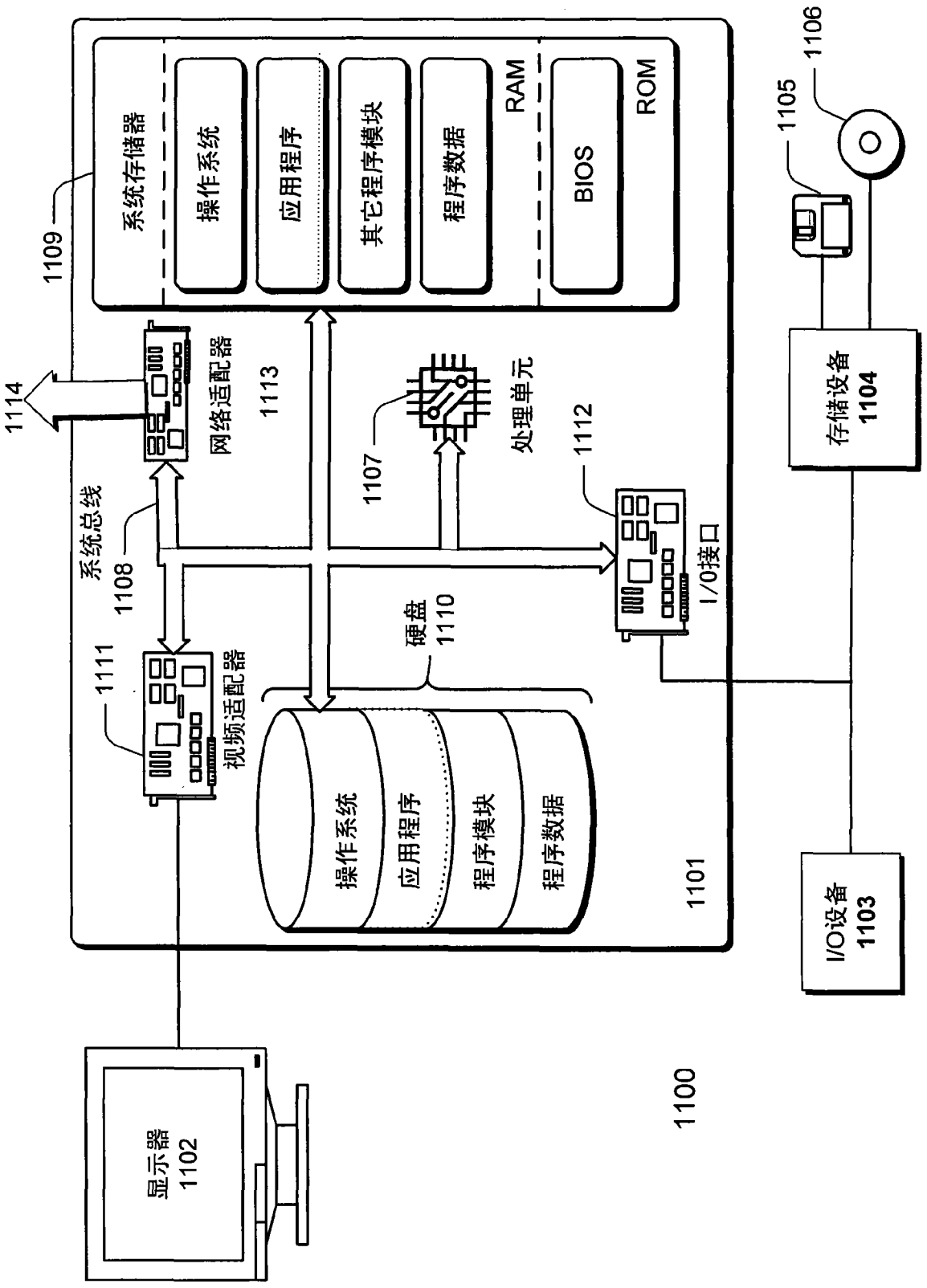

1100
显示器
1102
1101
系统存储器
1109
操作系统
应用程序
其它程序模块
程序数据
RAM
BIOS
ROM
网络适配器
1113
1114
处理单元
1112
1107
系统总线
1108
I/O接口
硬盘
1110
视频适配器
1111
操作系统
应用程序
程序模块
程序数据
存储设备
1104
1105
1106
I/O设备
1103

图 11