(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102227718 A
(43)申请公布日 2011.10.26

(21)申请号 200980147966.5

(22)申请日 2009.10.30

(30)优先权数据
12/324,305 2008.11.26 US

(85)PCT申请进入国家阶段日
2011.05.26

(86)PCT申请的申请数据
PCT/US2009/062746 2009.10.30

(87)PCT申请的公布数据
WO2010/062679 EN 2010.06.03

(71)申请人 微软公司
地址 美国华盛顿州

(72)发明人 N·斯里尼瓦桑 R·W·舒米德尔 N·阿布多

(74)专利代理机构 上海专利商标事务所有限公司 31100
代理人 顾嘉运

(51)Int.Cl.
G06F 15/16(2006.01)
G06F 13/14(2006.01)
G06F 9/06(2006.01)
H04L 29/06(2006.01)

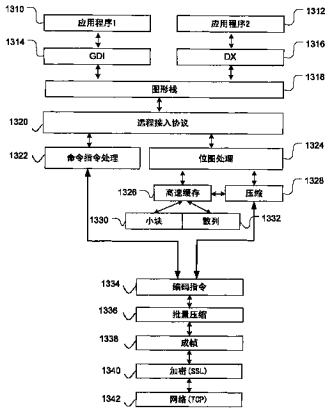
权利要求书 2 页 说明书 19 页 附图 16 页

(54)发明名称

用于远程桌面协议的硬件加速

(57)摘要

一种用于将远程终端服务处理任务卸载至一外围设备的方法，所述远程终端服务处理软任务否则将在计算机系统的处理器和存储器中执行。在一个实施例中，分层网络模型中利用所公开的方法，其中通常在网络应用程序中执行的计算任务改为卸载至诸如网络接口卡(NIC)之类的外围设备。



CN 102227718 A

1. 一种计算系统中的用于将远程接入处理任务卸载至外围设备 (504) 的方法，所述计算系统用于提供对终端服务器 (204) 或虚拟机 (121) 的远程接入，其中传输层 (500) 接口用于发送和接收远程接入数据单元，所述方法包括 ：

确定 (1402) 所述外围设备 (504) 包括用于实现一个或多个指定的远程接入操作任务的任务卸载能力 ；

向所述外围设备 (504) 发送 (1404) 所述外围设备将执行所述一个或多个操作任务的指示，包括将与后续数据单元一起使用的上下文信息 ；以及

使得 (1406) 所述一个或多个远程接入操作任务由所述外围设备 (504) 执行。

2. 如权利要求 1 所述的方法，其特征在于，所述一个或多个操作任务提供部分远程接入处理。

3. 如权利要求 1 所述的方法，其特征在于，所述一个或多个操作任务提供完整的远程接入处理。

4. 如权利要求 1 所述的方法，其特征在于，所述一个或多个操作任务包括位图压缩。

5. 如权利要求 1 所述的方法，其特征在于，所述一个或多个操作任务包括批量压缩。

6. 如权利要求 1 所述的方法，其特征在于，所述一个或多个操作任务包括高速缓存。

7. 如权利要求 1 所述的方法，其特征在于，所述操作任务根据所述计算机系统，网络连接类型和远程客户机中的至少一个的当前需求来被选择性地卸载至所述外围设备。

8. 如权利要求 1 所述的方法，其特征在于，还包括通过在关联于所述外围设备的任务卸载缓冲器中设置至少一个标记指示符启用任务卸载能力的动作。

9. 如权利要求 1 所述的方法，其特征在于，所述外围设备是网络接口卡。

10. 如权利要求 1 所述的方法，其特征在于，数据包跨分层网络模型传输，并且其中所述数据包包括网络数据和包扩展数据。

11. 如权利要求 10 所述的方法，其特征在于，所述数据包指示所述外围设备将执行一批操作任务，并且其中所述包扩展数据包括指示至少一个操作任务将由所述外围设备执行的至少一个数据字段。

12. 一种适用于将远程接入处理任务卸载至外围设备的系统 (100)，包括 ：

至少一个处理器 (160) ；以及

通信地耦合至所述至少一个处理器 (160) 的至少一个存储器，所述存储器中存储有计算机可执行指令，所述指令用于 ：

向外围设备 (504) 发送指示所述设备将执行远程接入协议操作任务的数据包，所述远程接入协议操作任务包括位图压缩、批量压缩或高速缓存中的一个 ；以及

当确定所述外围设备 (504) 不包括用于实现所述远程接入协议操作任务的任务卸载能力时，使得所述操作任务由所述处理器 (160) 执行。

13. 一种其上存储计算机可执行指令的计算机可读存储介质 (1700)，所述指令用于将远程接入处理任务卸载至外围设备，包括 ：

确定 (1710) 所述外围设备包括用于实现一个或多个指定的远程接入位图处理任务的任务卸载能力 ；

向所述外围设备发送 (1712) 所述外围设备将执行所述操作任务中的一个的指示，包括将与后续数据单元一起使用的上下文信息 ；以及

使得 (1714) 所述一个或多个远程接入操作任务由所述外围设备执行。

14. 如权利要求 13 所述的计算机可读存储介质，其特征在于，所述位图处理包括高速缓存和批量压缩。

15. 如权利要求 13 所述的计算机可读存储介质，其特征在于，所述任务卸载能力还包括多路复用 / 成帧、加密，和命令指令编码中的一个。

用于远程桌面协议的硬件加速

[0001] 背景

[0002] 由于诸如七层 ISO 模型或 Windows™ 操作系统使用的分层模型等分层体系结构，计算系统中的网络应用程序会需要大量的主机处理器资源。在各层之间传递的数据包所执行的功能可以是软件密集型的，而且会需要大量的处理器和存储器资源。另外，许多计算机硬件外设，如网络接口卡 (NICs)，其性能、效率以及总处理能力也在持续增长。此类计算机系统外设通常配备有专用的处理器和存储器，而且通常能够执行否则得由计算机系统处理器在软件中执行的高级和复杂的计算任务。例如，许多 NIC 能够在适当的网络层中独立执行否则得由 CPU 在软件中执行的任务，如校验和计算 / 验证 ；数据加密 / 解密 ；消息摘要计算 ；TCP 或 UDP 分割 ；接收侧包分类 ；防御拒绝服务攻击的包过滤 ；以及其他任务。由此，将该 CPU 密集型任务卸载至外围硬件设备是有益处的。这将减少处理器的使用和主机计算机中存储器带宽的使用，因而有利于提高整个系统的效率、速度和总处理能力。

[0003] 远程计算系统可以使用户能够接入远程计算系统提供的资源。远程计算系统的服务器可以执行程序并向客户机发出表示用户接口的信号，所述客户机可通过在符合诸如 TCP/IP 协议的通信协议的网络上发出信号来连接。可以向每一连接的客户机提供一会话，即，包括一组资源的执行环境。每一客户机可向服务器发送表示用户输入的信号，服务器会将用户输入应用于适当的会话中。所述客户机可以使用诸如远程桌面协议 (RDP) 的协议来连接服务器资源。诸如 RDP 等协议通常处理图形、诸如 USB 的设备通信、打印机键盘和鼠标，此外还有服务器和客户机之间的应用程序的虚拟通道。在典型的服务器配置中，终端服务器主机有数以百计的客户机会话。

[0004] 诸如 RDP 的协议支持不同的远程客户机机器性能。通常，来自服务器的图形数据需要编码、加密为图元形式或在服务器上渲染，而且所得到的位图需要压缩 / 加密并跨网络传输。编码、渲染和加密操作在性质上是高度计算性的，并需要很高的 CPU 使用率。因而使用上述技术将该 CPU 密集型任务卸载至外围硬件设备将是有益的。

[0005] 概述

[0006] 本发明涉及用于将之前在处理器软件层执行的远程终端接入功能和任务卸载至耦合至计算机系统的适当的外围硬件的系统和方法。在一实施例中，可将任务卸载至网络接口卡 (NIC) 外围设备，该外围设备可执行否则由计算机 CPU 在软件中执行的任务中的部分或全部。

[0007] 在一实施例中，操作系统 (OS) 可以“查询”与计算机系统连接的外围硬件 (例如 NIC) 的设备驱动程序。不同的设备驱动程序可以通过标识它们各自的外围硬件的处理能力来响应，此处将该处理能力称为“任务卸载能力”。在一实施例中，一旦标识每一特定外围设备的任务卸载能力，OS 然后可使所选外围设备能够执行某些任务。OS 之后可请求外围设备在动态的按需的基础上执行之前启用的任务。

[0008] 在各个实施例中，当 RDP 协议处理过程中需要渲染和压缩功能时，这些任务可以包括渲染和 / 或压缩。因而网络接口卡的硬件能力可以通过使用诸如 TCP/IP 等于协议数据流内联的其他已卸载的网络卸载任务之上的 RDP 级处理的卸载路径来利用。

[0009] 除前述的之外，在构成本发明的一部分的权利要求书、附图，以及文本中描述了其他方面。本领域技术人员将理解，本发明的一个或多个方面可包括但不限于用于实现本发明的本文所述方面的电路和 / 或编程 ；该电路和 / 或编程实质上可以是配置成实现本文所述方面的硬件、软件和 / 或固件的任何组合，这取决于系统设计者的设计选择。

[0010] 以上是概述，并且因此必然包含细节的简化、一般化及省略。本领域技术人员将明白，本概述只是说明性的并且决不旨在是限制性的。

[0011] 附图简述

[0012] 图 1 描绘其中可实现本发明的各方面的示例计算机系统。

[0013] 图 1a 示出了具有包括多个虚拟处理器以及对应的客操作系统的多个虚拟机的虚拟机环境 ；虚拟机由可包括调度器和其他组件的虚拟化层来维护，其中虚拟化层虚拟化多个虚拟机的硬件 ；

[0014] 图 2-4 描绘了用于实施本发明的各方面的操作环境。

[0015] 图 5 示出了此处所公开的网络栈的功能层和旁路路径。

[0016] 图 6 示出了此处所公开的 NDIS 路径和旁路路径的功能层。

[0017] 图 7 示出了例示此处所公开的卸载机制的梯形图。

[0018] 图 8 是示出主计算机和外围设备之间的同步的框图。

[0019] 图 9 示出了存在于分层网络体系结构中的某些功能组件。

[0020] 图 10 示出了描述根据本发明的通过程序组件的数据包流的功能框图。

[0021] 图 11 示出了数据包和包扩展的实施例。

[0022] 图 12 示出了描述远程终端服务处理的示例体系结构。

[0023] 图 13 是示出远程终端服务处理的各方面的功能框图。

[0024] 图 14 描绘了用于实施本公开的各方面的示例性操作过程。

[0025] 图 15 描绘了用于实施本公开的各方面的示例性操作过程。

[0026] 图 16 描绘了用于实施本公开的各方面的示例性操作过程。

[0027] 图 17 示出了承载参考上面的图 1-16 所讨论的计算机可执行指令的计算机可读介质。

[0028] 详细描述

[0029] 概括的计算环境

[0030] 在以下描述和附图中阐明了某些具体细节，以提供对本发明的各个实施例的全面理解。通常与计算和软件技术相关联的某些公知细节将不在以下公开中描述，以避免不必要地使当前公开的主题的各实施例晦涩难懂。此外，相关领域的普通技术人员可以理解，他们可以无需以下描述的细节中的一个或多个而实现当前公开的主题的其它实施例。最后，尽管在下面的公开中参考步骤和序列描述了各种方法，但是，如此的描述用于提供当前公开的主题的各实施例的清楚的实现，且步骤和步骤的序列不应该理解为实施本发明所必需的。

[0031] 应该理解，此处所描述的各种技术可以结合硬件或软件或，在适当的情况下，结合两者的组合来实现。因此，本发明的方法和装置或其某些方面或部分，可以采用包含在诸如软盘、CD-ROM、硬盘驱动器或任何其它机器可读存储介质等有形介质中的程序代码（即，指令）的形式，其中，当程序代码被上载至诸如计算机等机器并由其运行时，该机器成为用于

实现所公开的主题的装置。在程序代码在可编程计算机上执行的情况下，计算设备一般包括处理器、可由处理器读取的存储介质（包括易失性和非易失性存储器和/或存储元件）、至少一个输入设备，以及至少一个输出设备。一个或多个程序可以例如，通过使用API、可重用控件等来实现或利用结合当前公开的主题描述的过程。这样的程序优选地以高级别过程或面向对象编程语言来实现，以与计算机系统进行通信。然而，若有需要，程序也可以以汇编或机器语言来实现。在任一情况下，语言都可以是编译的或解释的语言，并与硬件实现相结合。

[0032] 远程桌面系统是维护可由客户计算机系统远程地执行的应用的计算机系统。输入是在客户计算机系统处被输入的，并通过网络（例如，使用基于国际电信联盟(ITU)T.120协议家族等协议，如远程桌面协议(RDP)）传送到终端服务器上的应用。该应用如同该输入是在终端服务器处送入的那样来处理该输入。该应用响应于所接收到的输入生成输出，并且该输出通过网络传送到客户计算机系统。客户计算机系统呈现输出数据。由此，在客户计算机系统处接收输入并呈现输出，而处理实际上是在终端服务器处发生的。会话可包括诸如桌面之类的外壳和用户界面、跟踪该桌面内的鼠标移动的子系统、将图标上的鼠标点击转换成实现程序实例的命令的子系统等等。在另一示例实施例中，会话可包括应用。在该示例中，当呈现应用时，桌面环境仍可被生成并对用户隐藏。应当理解，前述讨论是示例性的，且当前公开的主题可以在各种客户机/服务器环境中实现且不限于特定终端服务产品。

[0033] 即使不是全部，也是在大多数远程桌面环境中，输入数据（在客户计算机系统处送入的）通常包括表示对应用的命令的鼠标和键盘数据，且输出数据（由终端服务器处的应用生成）通常包括用于在视频输出设备上显示的视频数据。许多远程桌面环境也包括扩展到传输其他类型的数据的功能。

[0034] 可使用通信信道来通过允许插件经由RDP连接传输数据来扩展RDP协议。存在许多这样的扩展。诸如打印机重定向、剪贴板重定向、端口重定向等特征使用通信信道技术。由此，除了输入和输出数据之外，可以有许多需要来传输数据的通信信道。因此，可能有传输输出数据的偶然请求和传输其他数据的一个或多个信道请求争用可用的网络带宽。

[0035] 图2示出了启用终端服务的实现200。TS客户机202和TS 204使用RDP来进行通信。TS客户机202运行TS客户机进程206，该TS客户机进程206向已经在TS上派生的TS会话210发送诸如例如键盘数据和鼠标点击数据之类的RDP输入设备数据208，并且接收诸如用户界面图形数据之类的RDP显示数据212。一般而言，TS客户机进程206是瘦客户机进程并且大多数处理在TS 204上提供。

[0036] 图3示出了通过防火墙302启用终端服务的实现300。远程TS客户机304通过网络308连接到终端服务网关(TSG)306。TS客户机上的超文本传输协议(HTTP)传输进程310以及TSG 306上的HTTP进程312方便通过防火墙302来进行通信。HTTP传输进程310将诸如远程过程调用(RPC)数据或RDP数据之类的数据包装在TSG 306的HTTPS首部中。TSG 306可经由套接字输出(socket out)进程316通过套接字连接318连接到TS 314。一旦TS客户机304得到认证并且建立连接，则可以在TS客户机304和TS 314之间来回传递RDP数据320。

[0037] 图4示出了实现400的一般化示例，其中利用现有远程过程调用/超文本传输协

议 (RPC/HTTP) 代理，由此经由防火墙 402 通过 RPC/HTTP 连接提供诸如 RDP 之类的终端服务协议。该实现的体系结构示出通过将 RDP 协议包装在 RPC 调用中，能够有利地利用现有基于 RPC 的代理。具体而言，TS 客户机 406 上的 RPC 传输插件 404 将提供 TS 客户机 406 和终端服务器 408 之间的通信的 RDP 流包装在 RPC 协议中。这方便利用基于 RPC 的代理，由此允许进行防火墙导航。可以在 TS 上以用户模式运行的基于 RPC 的代理 410 可将接收到的数据转发给可以在 TS 上以内核模式运行的套接字监听器 412。

[0038] 如上所讨论的，客户机可使用诸如远程桌面协议 (RDP) 等远程协议来连接到使用终端服务的资源。当远程桌面客户机经由终端服务器网关连接到终端服务器时，该网关可打开与终端服务器的套接字连接，并将客户机通信重定向到 RDP 端口或专用于远程访问服务的端口。网关还可使用通过 HTTPS 传送的终端服务器网关协议来执行与客户机的特定的网关专用交换。

[0039] 诸如管理程序等虚拟机监控程序是创建虚拟机的程序，每一个虚拟机带有可由底层物理硬件资源支持的虚拟化硬件资源。图 1a 示出了虚拟机环境 100，带有多个虚拟机 120、121，这些虚拟机包括多个虚拟处理器 110、112、114、116，以及对应的客操作系统 130，132。虚拟机 120、121 由虚拟化层 140 来维护，该虚拟化层 140 可以包括调度器 142 及其他组件（未示出），其中虚拟化层 140 为多个虚拟机 120、121 虚拟化硬件 150。多个虚拟处理器 110、112、114、116 可以是底层硬件物理处理器 160、162 的虚拟对应物。

[0040] 用于实现上文所提及的分区的所有这些方案只是示例性实现，且此处没有任何东西应该解释为将本公开限制为任何特定虚拟化方面。

[0041] 使用卸载的硬件加速

[0042] 本发明的实施例一般涉及提供减少处理单元 21 的处理开销以及存储器使用的能力。这主要通过将特定的计算任务卸载至与计算机系统 20 相连的适当的外围硬件设备来完成的，该卸载例如可以通过在处理单元 /CPU 21 上执行的操作系统、应用程序和 / 或其他程序模块来完成。许多外围设备逐渐配备有专用的处理器和存储器，并且完全能够执行许多同样的在典型情况下仅由 CPU 21 完成的任务。此类设备的例子可以包括，例如，网络接口卡（图 1 中的 53）；盘片驱动器接口卡（例如，图 1 中的 32，33，34）；小型计算机系统接口 (SCSI) 设备；智能串行接口卡；或者和相关与应用的外围设备，例如用于数据加密 / 解密的设备。

[0043] 尽管此处讨论的大致的发明概念可以用于将计算任务卸载至任何上述外围硬件设备，但所披露的主题将参考当前优选的实施例的例子来描述，其中计算任务被卸载至网络通信设备，例如图 1 中所示的 NIC 53。更具体地，所描述实施例被论作实现于在联网环境和微软公司提供的 Windows 操作系统体系结构中。但应当意识到的是，虽然具体参考的是 Windows 概念和技术，但本领域技术人员将认识到许多操作系统和连网体系结构，如果不是大多数的话，与当前披露中的环境都具有类似之处。

[0044] 图 5 示出了构成连网模型的组件与本发明的组件之间的相互关系。在正常的操作中，网络消息由应用程序 500 通过网络栈 502 发送至外围设备 504，其中该消息被发送至其他设备和应用程序并且是从其他设备和应用程序接收到的。网络栈 502 包括一个或多个中间软件层 506，中间软件层 506 对数据进行特定的操作，例如对数据打包，可靠数据传输，数据加密以及消息摘要计算。

[0045] 交换机 508 用于将为中间软件层 506 执行的网络栈操作从处理单元 150 中卸载。尽管交换机 508 是单独示出的，应当注意的是交换机 508 可以集成在网络栈 502 的最上面的中间层中。数据经由外围设备 504 的烟囱 (chimney)510 发送至外围设备 504 以执行网络栈操作。在该层次关系下，中间软件层不必独占地驻留在主机或外围设备中，而且允许任何中间层都可以完全卸载，或保持在主机中，或是二者的组合（例如，卸载一个或多个特定的连接）。此外，烟囱可以层叠在烟囱上（例如，IPSEC 烟囱可以层叠在 TCP 烟囱之上）。连接可以是可靠和非可靠数据传输及单播或多播数据传输的任意组合。如果中间层保持在主机中，则主机将对高速缓存在外围设备 504 中的变量（将在下面进行描述）进行更新。例如，可以将传输控制块 (TCB) 的连接状态项卸载至传输层，同时将网络层的路由高速缓存项 (RCE) 卸载至外围设备 504。交换机 508 在通过卸载的 TCB 的烟囱 510 发送通信时，继续通过共享同 RCE 的网络栈 502 为不同的 TCB 发送通信。

[0046] 交换机 508 通过向中间层 506 发送卸载请求来启动卸载。该卸载请求包括资源信息，该资源信息帮助外围设备 504 确定其是否能够成功地卸载连接。每一中间层 506 或者拒绝该卸载请求，或者在卸载请求中添加资源信息并将卸载请求发送至网络栈 502 中相邻的软件层。外围设备 504 在接收到该卸载请求时，计算器是否有可用的资源以卸载连接。如果卸载是不可能实现的，则外围设备 504 拒绝该卸载请求。否则，外围设备 504 接受该卸载请求并为该连接分配资源。外围设备 504 通过向中间软件层 506 发送具有参数链表的完成消息来完成该卸载请求。该参数链表向中间软件层 506 和交换机 508 提供信息，以允许中间软件层 506 和交换机 508 与外围设备通信。每一中间软件层 506 从该参数链表中移除该层的信息。

[0047] 当一中间层 506 接收到卸载操作的完成消息时，该中间层 506 将其状态传递至外围设备 504。每一状态可以有三种类型的变量 ;CONST、CACHED，和 DELEGATED。状态可以有全部三种类型的变量，或者也可以有三种类型的变量的子集。CONST 变量为在卸载的连接的整个生命周期中都不发生改变的常量。在该连接被上载时，它们并不会被读回至该层中。该主机处理单元 21 维持对 CACHED 变量的所有权，并确保主机处理单元 21 中 CACHED 变量的任何改变都会在外围设备 504 中进行更新。改变 CACHED 状态的控制消息由网络栈 502 处理。因此，当连接被上载时，主机将对 CACHED 变量进行写入但不需要读回。主机处理单元 21 将 DELEGATED 变量的所有权转移至外围设备 504。当卸载发生时 DELEGATED 变量被写入一次，当卸载终止时 DELEGATED 变量被读回。通过仅传回 DELEGATED 变量，将连接传回至主机的开销被最小化。必须共享于（例如，控制于）网络栈 502 和由于多种性能原因正在被卸载的（即，被委派的）外围设备 504 之间的状态被清楚地分割在网络栈 502 和烟囱 510（例如，TCP 卸载中的 IP ID) 之间，因而网络栈 502 和外围设备 504 各自拥有该状态中独占的一部分。当需要时（例如用于统计），主机处理单元 21 向外围设备 504 查询 DELEGATED 变量。主机处理单元 21 也可以查询 CONST 或 CACHED 变量以用于诊断。将状态划分为三类使得网络栈 502 和烟囱 510 能够清楚地共存。应当说明的是，可以将该状态包括在卸载请求中。该状态不包含委派的状态变量时，或者包含在初始卸载请求和卸载请求完成时并没有改变的委派的状态变量时，都可以进行上述操作。

[0048] 外围设备 504 或主机决定卸载的连接何时将被上载。该上载可以由外围设备 504 或交换机 508 启动。一旦上载启动，外围设备 504 完成全部未完成的具有适当状态的请求，

并将最上面的中间层的委派的状态移交至交换机 508。交换机 508 对后续的传输请求进行排队，并停止公布接收缓冲器。交换机 508 指示最上面的中间层控制该委派的状态。最上面的中间层控制该委派的状态，并向交换机 508 发送一完成消息。在接收到完成消息后，交换机 508 确认向外围设备 504 的上载，使得外围设备 504 释放不再使用的资源。

[0049] 需要注意的是，该最上面的中间层将卸载的连接的输入数据包转发至外围设备 504，该外围设备 504 是为处理单元控制委派的状态的外围设备。数据包可以在外围设备将委派的状态移交至交换机 508 的时刻与最上面的中间层控制该委派的状态的时刻之间到达。在将委派的变量移交至交换机 508 之后，外围设备 504 就不能再处理输入数据包。当外围设备 504 接收到输入数据时向该最上面的中间层发送一错误消息，指示上载操作在进行中。该错误消息通知该最上面的中间层停止转发输入数据，并对后续的数据进行缓冲，直至该最上面的中间层接收到委派的状态。或者，以外围设备 504 上额外的缓冲存储器为代价，输入数据可以转发至外围设备 504 以供进行缓冲。

[0050] 中间软件层 506 可以将多个连接卸载至外围设备 504。中间软件层 506 维持一上层状态对象（即在该中间软件层 506 之上的层的状态对象）数量的引用计数器，该上层状态对象引用用于卸载的中间软件层的状态对象。此处所使用的状态对象为特定层的状态变量的集合，此处分类为 CONST、CACHED，或 DELEGATED。如果一中间层的卸载状态对象并没有上层指向它的引用，则中间层 506 向外围设备 504 发送消息，以上载该中间层的状态对象并向中间层 506 发送委派的状态变量。外围设备 504 删除中间层 506 的状态对象，中间层 506 向交换机 508 发送完成消息。

[0051] 图 9 示出了构成 Windows 上下文中的网络模型的部分组件的简化图。出于说明的目的，对应于多种 Windows 组件的 OSI 层也被示出。在底层中，对应于 OSI 模型中的物理层，驻留有实际的 NIC（有时也称作网卡，或网络适配器）900-904。NIC 为硬件设备，根据特定的网络拓扑结构提供与物理介质（网络电缆）间的物理连接以及信号的传输，该信号携带有全部更高的层所产生的数据。如上所述，很多 NIC 配备有专用的处理器和存储器，并能够执行额外的复杂计算任务—包括否则得由主机处理器处理的任务。NIC 物理上可以由设置在计算机插槽中的印刷电路板卡来实现，如设置于遵守 PCMCIA 的插槽中的 PCMCIA 型卡，如设置于主板上计算机底架中的专用芯片，或其他适当的形式。

[0052] 每一 NIC 经由一对应的网络驱动程序 916-920 与 Windows 网络模型逻辑上互联，图中用双向线 908-912 示意性的表示。网络驱动程序驻留在网络模型的 MAC 子层中，经由对应的 NIC 将 Windows 与物理网络通道相连。每一驱动程序，通常由对应的 NIC 的供应商提供的软件组件来实现，负责通过它对应的网络连接发送和接收包并代表操作系统管理 NIC。每一驱动程序在对应的 NIC 上还开启 I/O 并从中接收中断，并向上调用协议驱动程序以通知它们向外的数据传输的完成。同样地，设备驱动程序也负责对应的 NIC 中额外的处理能力的调用、控制和 / 或监控。

[0053] 在某些环境中该驱动程序组件被编写为实现单独的特定网络协议，例如 TCP/IP 或 XNS。当前公开的主题可以应用于这样的环境中。但是为了具体地说明，本发明连同 Windows 网络体系结构一起进行描述，该网络体系结构中提供了称作网络驱动程序接口规范 (NDIS) 的接口和环境。该 NDIS 接口在图 9 中以 926 功能性地示出。NDIS 对每一网络驱动程序 916-920 屏蔽不同的传输协议（其例子以 928-934 示出）的细节，反之亦然。更具体地，

NDIS 描述了一个或多个 NIC 驱动程序 (916-920) 通过它与一个或多个底层 NIC(900-904)、一个或多个上层的传输协议驱动程序、或传输（图 9 中以 928-934 表示），以及操作系统之间进行通信的接口。

[0054] 本质上，NDIS 为 NIC 驱动程序开发定义了一完全抽象的环境。因而，对于 NIC 驱动程序需要执行的每一外部功能，从注册和截取 NIC 硬件中断到与传输协议驱动程序通信到经由寄存器操作和 I/O 端口与下层的 NIC 通信，都可以依靠 NDISAPI 来执行该功能。为了提供该级别的抽象以及由此带来的可移植性，NDIS 使用一导出库，称为 NDIS 接口库包装器（未示出）。NIC 驱动程序和协议驱动程序，NIC 驱动程序和操作系统，以及 NIC 驱动程序和 NIC 之间的全部相互作用都通过调用包装器函数来执行。因而，网络供应商提供 NDIS 接口作为网络驱动程序的最上层，而不是为 Windows NT 编写一具体传输的驱动程序。此做法允许任何协议驱动程序通过调用该接口将网络请求重定向至网卡。因而，用户可以使用一块网卡和一单独的网络驱动程序来通过 TCP/IP 网络和 DLC（或 NWLINK，或 DECnet，VINES，NetBEUI 等）网络进行通信。

[0055] 在网络和数据链路层上的是传输、协议和相关的驱动程序，在图 9 中以 928-934 示例性的示出。在 Windows 中，传输协议驱动程序是实现传输驱动程序接口 (TDI) 或可能在其上边的其他具体应用接口的软件组件，以向网络用户提供服务。在 Windows 中，该 TDI 为在会话层通信的网络组件提供一公共接口，如功能块 938 示出的重定向器和服务器功能。如公知的，传输协议用作网络的数据组织者，实质上定义了数据如何向下一接收层呈现以及相应的数据打包过程。它们分配包（在 Windows 上下文中有时也称为 NDIS 包），把数据从发送应用程序复制到包中，并通过调用 NDIS 将包发送至更低层的设备驱动程序，因而数据可以经由对应的 NIC 发送至网络上。

[0056] 应当意识到的是，额外的功能或任务也可以当该数据包通过不同网络层时在数据包上执行，通常在网络模型的第 3 层和第 4 层。根据本发明，上述额外的功能或任务可以改为由 NIC 硬件执行。

[0057] 例如，一个传统的由传输协议驱动程序执行的任务可以是计算校验和值然后将其附加至包中。这有助于数据在遍历网络链接时保证其完整性。一般地，该操作要求与网络包的发送方对应的传输协议添加一数字，该数字是通过把组成包的数据元素相加计算后所得的。然后包的接收方将添加的校验和数字与该数据对比，从而确定数据在传输中没有改变。上述校验和计算和对比可以改为卸载至 NIC 硬件。

[0058] 最好可以由 NIC 硬件执行的另一相关的任务为数据包的消息摘要计算。和校验和类似的，消息摘要用于保证包中数据的完整性。另外，消息摘要可以用于确认发送消息的一方为其所声称的，以此来保证数据的真实性。消息摘要的计算是非常 CPU 密集型的，而且是使用软件实现的话开销较大的功能。

[0059] 另一可取的功能为包中数据的加密。加密指的是包中消息的加密转换过程，因而未经授权的包的读者预先不知道密钥的话实际上不可能看到消息的内容。当然，加密算法也往往是 CPU 和存储器密集型的，如果采用软件实现的话开销非常之大。此类加密的例子包括安全套接字层协议 (SSL) 加密和互联网安全协议加密即“IPSec”。如公知的，SSL 和 IPSec 加密协议都是非常 CPU 和存储器密集型的。

[0060] 另一可以在数据包上执行的任务为 TCP 或 UDP 分割。如公知的，TCP 和 UDP 协议

将大的数据包分割成各段，这些段按照底层网络允许的最大的数据尺寸分割。例如，以太网允许网络上最大的包为 1514 字节。因而，如果 TCP 或 UDP 必须发送 64K 字节，则必须将数据分解为 1514 字节的段。

[0061] 此外，在从网络接收数据包时，经常在数据包上执行包分类。包分类包括用于服务质量 (QoS) 的数据包鉴别。换言之，每一数据包包含定义服务模式的字段，该服务模式应当在数据包上执行以保证最优性能。例如，包括视频和音频的数据包可能需要执行特定的功能从而在视频和音频呈现时保证高保真度。根据本发明，可以将为了鉴别服务模式的数据包分类卸载至 NIC 硬件。

[0062] 包过滤也可以由 NIC 硬件而不是 CPU 来执行。具体地，可以对数据包进行评估以确定它们是否具有作为拒绝服务攻击的一部分的特征。经常地，服务器软件是在请求服务的客户机为非恶意的假定下创建的。但是，受过充分训练的但是恶意的客户机用户可能会向服务器发出请求，该请求设计为阻止服务器为其他用户服务。拒绝服务攻击过滤指的是评估数据包以确定其是否具有拒绝服务攻击的特征的能力。

[0063] 拒绝服务攻击的一个例子是“SYN 洪泛攻击”，其中，由于一个或其他原因，在握手序列中，客户机不向服务器的同步 - 确认 (SYN-ACK) 响应发出任何最终应答。这导致服务器持续发出信号直至服务器最后超时。

[0064] 拒绝服务攻击的另一类型称为碎片或“泪珠”攻击。互联网协议要求对包进行分割，如果该包在下一路由器处理器时太大的话。后续的分割后的包确定一相对于第一包的开始的偏移，使得整个包在接收端能够被重新组装。在碎片攻击中，用户在后续的片段中放入一伪造的偏移值，经常导致接收端功能障碍。该 NIC 硬件可以配置为通过评估数据包来确定它们是否具有特定种类的拒绝服务攻击的特征，以过滤此类拒绝服务攻击。

[0065] 此类功能以及其他功能通常由计算机 CPU 20 在驻留在各个网络层中的软件组件中执行，因而会使用大量计算机资源，导致计算机系统性能的整体下降。因此，对此类或其他类似的任务进行卸载，从而可以在对应的 NIC 硬件执行它们，能够显著提高计算机系统的整体速度和效率。

[0066] 参考图 6，所示出的实施例中，外围设备 504 为 NIC 56，交换机 508 为传输层接口交换机 (TLI)606，网络栈 502 包括传输层 600，网络层 602 和成帧层 604。网络层 602 也被认为是路径层，该成帧层 604 也被认为是邻接层。

[0067] 网络消息在操作过程中由应用程序 500 通过网络栈 502 发送至 NIC 56。应用程序 500 发出的数据穿过 TLI 交换机 606，该 TLI 交换机 606 控制数据下行到基于主机的网络栈 502 或烟囱 608。要注意的是该 TLI 交换机 606 可以合并至网络栈 502 的顶层中。网络栈 502 的软件层从应用程序 500 接收数据，将其打包成包形式，并经由 NDIS 微型驱动程序 610 发送至外围设备硬件 614。数据包在栈 502 传输时，网络栈 502 需要执行的其他任务包括数据加密，可靠的数据传输，以及消息摘要计算 (例如，数据包的校验和或 CRC)。这些任务中的很多都由处理单元 21 执行，而且是处理器密集型的。

[0068] TLI 交换机 606 通过把连接的数据经由烟囱 608(和烟囱驱动程序 612) 发送至 NIC 56，将栈操作的执行从处理单元 21 上卸载。本领域技术人员应当明白的是，NDIS 微型驱动程序 610 和烟囱驱动程序 612 为微软 RTM 操作系统中的 NDIS API。为了清楚的解释，基于传输控制协议 (TCP) 的协议栈将用于解释所公开的主题。但是，应当体会到的是，本领域

技术人员可以明白，在当前公开的教导下，可以使用多种类型的外围设备，被卸载的可以是其他网络栈。例如，被卸载的可以是基于流控制传输协议 (SCTP) 或用户数据报协议 (UDP) 的协议栈。此外，本发明也可以用于卸载高级功能协议，例如因特网小型计算机系统接口 (iSCSI)，网络文件系统 (NFS)，或公共接口文件系统 (CIFS)。

[0069] 卸载操作发生的原因有很多。下文示例性而并非限制性地提供了多种原因中的一些。系统管理员可以选择一特定的服务来卸载。可以将一特定的连接卸载，如果该连接上的通信（按照字节或包的数量来计）正在消耗大量的资源。多种类型的服务都可以被卸载。例如，被卸载的可以是诸如 IPSEC 的安全协议。可以按照策略来驱动卸载操作。例如，一管理员的策略可以是首先卸载来自一组织范围内的连接。正在使用的系统资源（例如，cpu 的使用，数据高速缓存的使用，页表高速缓存的使用，存储器带宽）可以引起主机处理器卸载连接。

[0070] 图 7 示出了卸载 TCP 连接所执行的步骤。使用的是三步法。一般地，该三步法为分配卸载 TCP 连接所需的资源，向层 500，502，504 和 506 中的每一层提供句柄，以及将层 500，502，504 和 506 中每一层的状态卸载至 NIC 53。在卸载转换过程中，TLI 交换机 506 对应用程序 200 发出的全部消息进行缓冲。或者，传输层 500 对该数据进行缓冲。当该卸载操作完成时，被缓冲的数据传输至 NIC53，所使用的机制与卸载操作的数据传输相同。在卸载转换过程中当接收到输入包时，NIC 53 继续通过层 500，502，504，506 向上搬移数据，直到传输层委派的状态移交至 NIC 53。

[0071] TLI 交换机 506 通过向传输层 500 发送卸载请求（线条 700）来发起卸载操作。该卸载请求包括指向下一层的本地状态的指针（例如，对传输层 500 为一 TCB 指针，对网络层 502 为 RCE 指针，对成帧层 504 为 ARP 表指针，或对 NDIS 微型驱动程序 510 为 NDIS 微型端口指针），卸载类型（例如，对 TLI 交换机 506 为 TCP，对网络层 502 为 IPv6），以及帮助 NIC 53 确定其是否能够成功卸载该 TCP 连接的资源信息。TLI 交换机 506 还可以向 NIC 53 提供分派表。传输层 500 或者拒绝该卸载请求，或者向网络层 502 发送一卸载请求并将 TCP 资源信息添加至 TLI 交换机资源信息（线条 702）。

[0072] 网络层 502 接收该卸载请求，或者拒绝卸载该连接，或者向成帧层 504 发送一卸载请求并将网络资源需求添加至该 TCP 资源信息以及 TLI 交换机资源信息（线条 704）。网络层 502 还可以向 NIC53 提供分派表。成帧层 504 或者拒绝卸载该连接，或者向 NIC53 发送一卸载请求并将成帧资源需求添加至网络资源需求，TCP 资源信息和 TLI 交换机资源信息（线条 506）。

[0073] NIC 53 接收该卸载请求，并计算是否有可用的资源来卸载该 TCP 连接。如果 NIC 确定该卸载操作是不可能的，则拒绝该卸载请求。如果 NIC 确定该卸载操作是可能的，则接受该卸载请求并为该连接分配资源（例如，TCB，路由高速缓存项 (RCE)，地址解析协议 (ARP) 表项 (ATE)）。NIC 53 创建参数链表和分派表以向层 500、502、504 和 506 移交，并通过向成帧层 504 发送一完成消息来完成该卸载请求（线条 408），该完成消息包括参数链表。该参数包括层 500、502、504 和 506 中每一层的卸载句柄和分派表。此处所使用的句柄指的是允许软件层与外围设备通信的机制。示例性而并非限制性地，卸载句柄可以是一基于指针的句柄，一用作数组查找的整数值，散列表（例如，散列函数），软件层（或网络栈）与外围设备之间的通信通道，或软件层向下传递的、外围设备用于查找状态对象的一组参数。

[0074] 分派表用于直接向 NIC 53 发送数据，或从 NIC 53 直接接收数据。分派表也可以用于提供诊断。例如，可以添加一软件层以对系统进行监控并注入错误来保证系统正确地运行。此外，分派表可以由软件层来修补，如果需要的话能增加额外的功能。例如，可以增加一软件层来提供过滤驱动程序的功能。修补过程通常通过以下方式实现：抓取指向插入所增加的函数处的原始函数的指针并将其重定向（例如指向）至增加的函数。在补丁被插入后，增加的函数执行其功能并且在原始函数被调用时对该原始函数进行调用。

[0075] 成帧层 504 将给成帧层的卸载句柄和分派表存储在它的 ARP 表项中，从而在目的 MAC 地址改变或封装类型改变时能简单地更新。成帧层 504 然后更新与 ATE 相关的 NIC 53 状态（线条 710）。成帧层 504 从该链表移除它的状态，并将该链表中剩余的信息转发至网络层 502（线条 712）。

[0076] 网络层 502 存储网络层 502 的卸载句柄和分派表。网络层 502 还把它的状态发送至 NIC53（线条 714）。网络层 502 从该链表中移除网络层信息，并向传输层 500 发送一完成消息（线条 716），该完成消息包括该参数链表和分派表。网络层 502 可以将为被卸载的状态接收的 IP 片段转发至 NIC53 进行处理，或者可以在网络层中处理该 IP 片段并将其转发至传输层 500。

[0077] 在其他的实施例中，层的状态对象与该卸载请求一起发送。例如，该成帧层状态对象和网络层状态对象与该卸载请求一起发送，而且只有该高速缓存的状态在卸载请求和完成事件之间发生了改变，该状态才会被更新。如果该委派的状态不是当前的或者在卸载请求及其完成之间无法改变，则全部层状态对象仅能和卸载对象一起发送。但是，分类为 CONST 的状态变量可以和该卸载请求一起发送，即使该委派的状态是当前的而且在该卸载请求及其完成之间可能发生改变。

[0078] 传输层 500 存储给传输层的卸载句柄，并把它的状态发送至 NIC 53（线条 718）。如果有未完成的发送或接收缓冲待解决，则传输层 500 将该缓冲返回至 TLI 交换机 506。一旦该传输层开始将缓冲交回至 TLI 交换机 506，TLI 交换机 506 就会停止向传输层 500 发送缓冲并对其进行排队，等待传输层 500 向 TLI 交换机 204 发送包括该参数链表和分派表的完成消息。传输层 500 返回全部缓冲，然后发送完成消息（线条 720）。一旦 TLI 交换机 506 接收到该完成消息，TLI 交换机 506 将该发送和接收缓冲转移至 NIC 53（线条 722）。TLI 交换机 506 使用分派表来公布全部未完成的和之后的接收缓冲，并发送至 NIC 53 来处理。在完成该卸载请求花费的时间过程中，每一层 500、502、504 或者拒绝新的对卸载的状态对象（也即和一层相关的状态对象）的卸载请求，或者对它们进行排队，直到卸载操作完成。

[0079] 如果该传输状态未被卸载至 NIC 53，传输层 500 仍然能够处理输入的 TCB 数据并将该数据移交至 TLI 交换机 506。如果 TCB 数据在一卸载操作中途到达，传输层 500 或者对该数据进行保持，或者对该数据进行处理并移交至 TLI 交换机 506。在传输层 500 将其状态发送至 NIC 53（线条 718）的时刻和 TLI 交换机将缓冲转移至 NIC 53（线条 722）的时刻之间，通过网络栈 202 进入的输入 TCB 数据被发送至 NIC 53。

[0080] 对随后的卸载请求，网络层 502 和成帧层 504 将从 NIC 53 接收到的之前的卸载操作的卸载句柄传递至 NIC 53。这通知了 NIC 53 给网络层 502 和成帧层 504 的资源已经分配，从而保存了 NIC 资源并加速了卸载操作。

[0081] 如前所述，层 500、502、504 将它们的状态传递至 NIC 53。每一状态具有三种类型

的变量 :CONST，CACHED 和 DELEGATED。CONST 变量是在卸载的连接的生命周期中从不会发生改变的常量。当连接终止时它们并不会被读回。主机处理单元 21 保有 CACHED 变量的所有权，并确保在主机处理单元 21 中对 CACHED 变量的任何改变都在 NIC 53 中进行更新。所以，主机会写入但从不会读回 CACHED 变量（除非系统诊断请求它这么做）。主机处理单元 21 将 DELEGATED 变量的所有权转移至 NIC 53。DELEGATED 变量在卸载操作发生时被写入一次，在该卸载操作终止时被读回。通过将 DELEGATED 变量传回，将连接传递回主机的开销被最小化。主机处理单元 21 在需要时向 NIC 53 查询 DELEGATED 变量。

[0082] 传输层 500 的 CONST 变量包括目的地端口，源端口，表示移动 IP 的标记，SEND 和 RECV 窗口比例因数，以及网络层 502 的 NIC 句柄，其中移动 IP 是“转交”(care-of) 地址会发生改变的情形。传输层 500 的 CACHED 变量为 TCP 变量和 IP 变量。TCP 变量包括有效 MSS，由 NIC 53 指示的接收中需复制的字节数，关闭 Nagling 的标记，表示需要保持活动的标记，以及保持活动设置（也即，间隔，探针数量，以及增量）。IP 变量包括 TOS 和 TTL。DELEGATED 变量包括当前的 TCP 状态，下一 RECV（也即，RCV.NEXT）的序号，接收窗口尺寸 (RCV.WND)，第一未确认数据 (SND.UNA) 的序号，下一 SEND(SND.NEXT) 的序号，曾发送的最大序号 (SND.MAX)，最大发送窗口 (MAX_WIN)，当前阻塞窗口 (CWIN)，慢启动阈值 (SSTHRESH)，平滑 RTT(8*A)，增量 (8*D)，当前转发数，下一转发的剩余时间，以及待回应的时间戳。

[0083] 网络层 502 的 CONST 变量包括目的 IP 地址 (IPv4 或 IPv6) 和源目的 IP 地址 (IPv4 或 IPv6)。网络层 502 的 CACHED 变量包括给成帧层 504 的 NIC 句柄。网络层 502 的 DELEGATED 变量包括 IP 包标识开始值。成帧层 504 的 CACHED 变量包括 ARP 地址和表示首部（例如，LLC/SNAP[Logical Link Control/Sub-Network Access Protocol] 或 DIX[Digital, Intel, Xerox]）格式的标记。

[0084] 由于网络层状态可以在多个连接之间共享，成帧层状态可以在多个路径（例如，IP 别名）间共享，因而该传输层状态包括网络层的句柄，该网络层状态包括成帧状态的句柄。维持该层次关系是有若干原因的。因为 IP ID 的命名空间必须在全部卸载的连接中以每一路径为基础进行管理，因此一连接需要一网络层的 NIC 句柄。因为路由更新可能会改变下一跳的地址而指向一新的 MAC 地址，因此一路径需要一成帧层的 NIC 句柄。该层次关系还压缩了 NIC 需要维持的状态的数量。例如，一 IPv4 的 ARP 更新可能会改变从一 IP 地址至一 MAC 地址的映射（例如，一接口在服务器上发生错误）。主机将 MAC 地址维持作高速缓存的变量，因而它仅需要对高速缓存的变量进行一单独的更新，全部连接在新的接口上都发生错误。

[0085] 一旦 TCP 连接被卸载，NIC 53 就负责为它发送的分配包标识符（例如，IP ID)。IP ID 或者按每一接口来卸载，或者按每一层状态对象来卸载。NIC 53 被分配了 IP ID 命名空间的一部分。在一实施例中，当网络状态被传递至 NIC 53 时，NIC 53 被分配了整个 IP ID 命名空间的一半，并被给予一 IP 包 ID 开始值来使用。NIC 53 使用下面的公式在它发送的 IP 包上产生 IP ID :Cur.sub.-IPID = [(Start.sub.-IPID_For_This_Path)+(Counter_For_This_Path)mo-d32k]mod 64K Counter_For_This_Path = Counter_For_This_Path+1。

[0086] 当卸载的连接被上载或无效时，NIC 53 把它将使用的下一个 IP ID 值转交至网络层来为发生的下一卸载操作进行存储，主机处理单元 21 继续使用它被分配的那部分 IP ID 命名空间。主机处理单元 21 可以使用全部 IP ID 命名空间，但是每次卸载操作发生时计数

器都得进行设置。

[0087] NIC 53 将数据按照接收的顺序放入接收缓冲器中，并按照它们公布给卸载连接的顺序来填充应用缓冲。许多应用程序在公布一接收缓冲前等待一接收指示。在一实施例中，NIC 53 具有全局缓冲池，该缓冲池用于一连接的数据到达且没有应用接收缓冲被公布的情形。该全局的缓冲池被用于整个卸载连接，并可以被用于实现 :1) 对乱序 TCP 传输的处理 ;2)IP 数据数据报的碎片整理 ;3) 缓冲复制算法而非零复制算法，如果应用程序公布的缓冲对零复制算法太小的话。或者，如果资源的有效使用不是需要关心的事，那么也可以使用按连接的缓冲池。但是，如果 NIC 不支持按连接的缓冲池或缺乏系统资源（例如，在存储器中没有足够的资源来锁定应用缓冲器）。

[0088] 现在参考图 8，一连接一旦被卸载至 NIC 53，到 NIC 53 就由两条路径。第一路径为通过 NDIS 微型驱动程序 510，通过成帧层 504，网络层 502 和传输层 500。第二路径为通过卸载的连接 808，该卸载的连接 808 被称为烟囱。从主机计算机的角度来看，在通信方面对于两条路径全部都是一样的。高速缓存的状态变量将该两条路径与处理单元 21 同步，如前所示地更新 NIC 53 中的高速缓存的状态变量。高速缓存的变量的更新用箭头 802、804 和 806 示出。

[0089] 当一输入的数据包到达时，NIC 53 确定输入的数据包是经过卸载的路径还是非卸载的路径（也即，通过 NDIS 微型驱动程序 510 和层 504、502、500 的 NDIS 路径）。在一实施例中，NIC 53 通过对源和目的 TCP 端口号，源和目的 IP 地址以及协议类型运行散列函数，来确定通过哪一路径来发送该输入的数据包。如果该散列与所卸载的连接的参数匹配（即，遍历散列桶链表，该连接的所有元组都精确匹配），就使用烟囱 808。如果该散列与散列索引不匹配，就使用通过网络栈 502 的非卸载的路径。更新高速缓存的状态的控制消息由主机处理。这使得 NIC 53 不必处理该卸载的连接以外的控制消息，如 ICMP、DNS 以及 RIP 消息。

[0090] 如前该，在 Windows 或类似的分层网络模型中数据传输的基本单位是数据包。在 Windows 环境中，该数据包被称作 NDIS 包。每一包从栈顶（即，ISO 栈中的层 5）移动至最低软件层（即，ISO 栈中的层 2）。因而，该包定义了数据在发送和接收中在各层行进时，每一等级中都一样的一数据结构。作为示例，图 10 示出了包向下穿过各层到达 NIC 时所遵循的路径，NIC 在 900 示为一以太网 NIC。如上所述，传输驱动程序 928 从一发送应用程序接收数据，并把它打包成符合底层协议的包形式，然后经由 NDIS 接口 926 将该包转发至较低等级的设备驱动程序 916。此外，该传输协议可以在包上执行其他功能（例如，校验和计算等）。或者，其他功能组件也可以驻留在网络层或数据链路层中，在该包上执行额外的功能，例如图 10 中示出的 IP 安全功能 1044（例如，加密和 / 或消息摘要计算）。

[0091] 在一实施例中，数据包 1042 为将计算任务卸载至诸如 NIC 硬件 900 的外围设备的工具。例如，图 10 中应用程序数据 1040 从网络模型的上层向下传递至一适当的传输协议驱动程序，比如 TCP/IP 928。该驱动程序将该数据重新打包为一适当的数据包 1042。然后，根据将要在该特定的数据包 1042 上执行的额外的功能，一功能组件被包括进来，将一预定义的称为包扩展的数据结构附加至该数据包中。该包扩展的内容指示数据包到达 NIC 53 时哪一或哪些任务将在数据包上执行，下文将进行更详细的讨论。当数据包 1042 到达网络驱动程序 916 时，驱动程序 916 查询包扩展的内容从而确定哪一任务（哪些任务）将由 NIC

53 执行。驱动程序 916 然后控制 / 操作 NIC 上的硬件，以使其执行通过该包扩展的内容所请求的功能任务。

[0092] 例如，在图 10 中，数据包 1042 传递至一软件组件 1044，可以单独来实现或作为传输协议驱动程序本身的一部分来实现，软件组件 1044 将一包扩展附加至包 1042 中。根据将被卸载的特定任务，数据被包括在包扩展中。例如，如果将实现的是一 IP 安全功能，那么指示 NIC 53 应当依照一指定的密钥对数据包进行加密的数据将被包括在其中。当然，软件组件 1044 可以附加预定义的数据以使多个功能中的任意一个，例如上文所讨论的那些，将在硬件级上执行而不是由驻留在网络层中的软件组件执行。设备驱动程序 916 将从该包扩展中提取信息，然后在 NIC 53 引用指定的任务。

[0093] 图 11 示出了数据包 1042 的大体结构的一个实施例。根据所使用的确切的网络环境，包 1042 可以是任何格式的，在 Windows 环境中包按照 NDIS 来格式化，并包括诸如包描述符的信息；意义由协作设备驱动程序和协议驱动程序来定义的标记；用于存储与该包相关的带外数据 (OOB) 的区域；和该包的长度有关的信息；以及指向和该包的数据内容有关的存储位置的指针。

[0094] 图 11 进一步示出了额外的数据结构字段——包扩展 1150，额外的数据结构字段附加至 NDIS 数据包来识别任务卸载操作。如之前讨论的，正是该包扩展 1150 定义了一数据结构，该数据结构包含了识别正在被卸载至目的 NIC 的特定任务所必需的信息。在一优选的实施例中，对每一任务卸载类型（例如，校验和，加密 / 解密，等等），一预定义的数据字段会被包括在包扩展 1150 中。该数据字段可以简单地为控制标记或标记的形式，仅指示被执行的特定功能（例如校验和），或者该信息也可以为指向一数据结构的指针的形式，该数据结构进一步定义了一任务应当如何完成。

[0095] NIC 53 可以配置为将包扩展 1150 中的任务卸载控制字段认为是仅应用于附属的包。因此，例如，如果一特定的包中包含一表示 NIC 53 将执行校验和运算的标记，那么 NIC 53 将仅在附属的包上执行校验和运算。但是，如果对于一给定的包没有该标记，那么 NIC 53 对该包不执行校验和运算。或者，包扩展 1150 中的任务卸载控制字段可以指明，该 NIC 将在该包和网络上发出的后续的包上执行卸载任务，直到 NIC 53 受到另外的指示。

[0096] 包扩展 1150 也可以指示在接收站的 NIC 站执行什么任务。例如，包扩展可以指示接收站 NIC 在从网络上接收到包时执行某些适当的功能，例如，解密，校验和运算，包分类，防御拒绝服务攻击的包过滤，包重新组装，以及任何其他的可以对 NIC 配置来执行的接收功能。当然，发送站可能并不知道接收站 NIC 的任务卸载能力。如果接收站 NIC 没有执行所请求的功能的能力，那么接收站 CPU 将代替实现该功能。因此，发送站可以对接收站 NIC 对一特定的包所做的操作具有一定的控制力。

[0097] 此外，发送站也可以在 NIC 53 接收包时使用包扩展来控制 NIC 53。例如，在网络上发出的一特定的包可以包括包扩展，该包扩展包括给 NIC 53 的指示，指示它在接收到包时，将对该包进行解密。在这个例子中，该指示并不应用在将在网络上发出的包上，因为该指令是为了对从网络上接收到的包执行功能。因此，在这个例子中，将在网络上发出的包的包扩展被用作一种机制，该机制用于控制在接收到包时，何种接收功能将被卸载至 NIC 53。发送站每发送一个包时，就有一次机会来控制 NIC 53 的接收功能。

[0098] 再次参考图 11 所示的例子，包扩展 1150 包含多个数据字段 1151 至 1153，该多个

数据字段 1151 至 1153 控制卸载至一 NIC(例如,NIC 53) 的发送功能,包扩展 1150 还包括多个数据字段 1154 至 1158,该多个数据字段 1154 至 1158 控制卸载至一 NIC(例如,NIC 53 或一接收站的 NIC) 的接收功能。

[0099] 例如,包扩展 1150 包括一表示 NIC 53 执行校验和运算的数据字段 115。这向发送 NIC 53 指出,发送 NIC 53 本身将在附加的包上执行校验和运算。

[0100] 包扩展 1150 也可以包括一安全功能数据字段 1152,来指示 NIC 53 应当执行安全功能,例如连同包数据的 SSL 或 IPSec 加密和 / 或消息摘要计算执行的安全功能。对于该类型的安全任务,字段 1152 优选地包括一指向包含一数据结构 (例如,数据结构 1160) 的存储位置,该数据结构依次包含与加密和 / 或消息摘要功能的执行过程相关的信息。在某些情况下,将指向具有相关数据的存储位置的指针包括进来,有利于在包扩展本身内存储实际的数据。

[0101] 参考图 11,校验和数据字段 1154 指示,对全部接收到的在发送站已经执行校验和计算的包,NIC 53 将执行校验和计算来确保数据没有在到达 NIC 53 的途中改变。校验和字段 1154 可以无限期地控制所有包的校验和接收功能,或者在一给定的时期内进行控制,除非随后的校验和字段明确地否定。

[0102] 该包扩展也可以包括一安全标记 1155,指示 NIC 53 在接收包时应当执行安全功能,例如连同包数据的 SSL 或 IPSec 解密和 / 或消息摘要计算执行的安全功能。对于该类型的安全任务,字段 1155 优选地包括一指向包含一数据结构 (例如,数据结构 1160) 的存储位置,改数据结构依次包含与加密和 / 或消息摘要功能相关的信息。

[0103] 该包扩展也可以包括一组装数据字段 1156,该组装数据字段 1156 指示 NIC 53 将接收到的包组装为批量数据,也可以包括一分类字段 1157,该分类字段 1157 指示 NIC 53 对每一包进行如上所述的服务质量的分类,也可以包括一拒绝服务 (DOS) 攻击过滤字段 1158,用于对输入的包过滤如上所述的 DOS 攻击的特征。包扩展 1150 也可以包括与标记 1154-1158 类似的用于接收站 NIC 在被攻击的包上执行的接收功能数据字段。

[0104] 在一实施例中,一特定的设备驱动程序对包扩展 1150 中包含的信息进行查询,其中,包 1142 被发送到该设备驱动程序。在所示实施例的 Windows 环境下,此类功能优选地通过进行适当的 NDIS 函数调用来执行。例如,可以进行一预定义的 NDIS 函数的调用,该 NDIS 函数给包返回一指向包扩展 1150 的存储位置的指针。然后设备驱动程序软件可以识别那些任务将被执行,并根据卸载的任务,以适当的方式运行 / 操作该驱动程序对应的 NIC 硬件。

[0105] 由于多种原因,利用实际的数据包来将计算任务从计算机处理器卸载至外围设备是有益的。例如,传输驱动程序可以一包接一包的方式来利用外围设备的性能。这使得任务被动态地下载,而且一外围设备的性能可以根据需要来使用。因此,如果在一特定的时间点计算机系统的处理开销较小,那最好以传统的方式在计算机处理器上执行某些任务。或者,如果 CPU 负担较重地上载了其他计算任务,那么它可以仅通过在数据包附加必要的包扩展来将任务卸载至外围设备。

[0106] 另一个好处是通过一个单独的包来卸载多个任务的能力,实质上是将多个操作一次成批处理。例如,当计算机处理器执行校验和运算或加密运算时,在该运算完成之前必须将整个数据字段装载至一存储位置,即包数据的校验和或加密计算完成之前。而且,由于分

层网络模型一次只能执行一项操作，因而需要多次把数据复制到存储器中。但是，按包进行的方法使得多个任务可以在一个包中被卸载。因此，外围硬件可以根据硬件能力在一次单独的数据传输中执行两项或更多项操作，因而显著地增大了计算机系统的总处理能力和效率。

[0107] 应当意识到的是，尽管上述方案在指定任务以将其卸载至一特定的 NIC 的能力方面具有显著的益处，但信息的按包传递也可以通过其他方式来使用。例如，如果一特定的 NIC 能够将包的送达控制在预定的时间，那么该包扩展数据结构可以用来传递用于确定 NIC 硬件应当怎样和 / 或何时发送包的信息。

[0108] 在一实施例中，在一传输协议驱动程序将一包扩展附加值一数据包以将一特定的任务卸载至 NIC 之前，将线执行两个额外的功能。在存在多种不同类型的硬件外围设备，每一种具有不同的处理能力的范围内，当前公开的实施例优选地提供了一种方法，通过该方法该传输驱动程序能首先查询连接至计算机系统的外围设备的任务卸载能力。一旦上述此能力被确定，传输协议驱动程序然后就能设置或使能所感兴趣的那些任务。一旦被使能，所指定的任务随后就能以上述方式在每一个包的基础上运用。

[0109] 远程接入协议的硬件加速

[0110] 如上提到的，远程接入和终端服务通常使用诸如远程桌面协议 (RDP，Remote Desktop Protocol) 的协议涉及桌面虚拟化。本领域技术人员应当意识到尽管当前公开按照 RDP 协议来进行描述，所公开的原理可以很容易地应用于提供远程接入服务的任何系统，例如虚拟网络计算 (VNC)，Citrix XenApp，以及相类似的。

[0111] 诸如 RDP 的协议设计为通过将图形显示信息从远程计算机传递至用户以及将输入从用户传输至远程计算机，该输入可能是在本地加入的，来推动用户和远程计算机的交互。该协议还提供一可扩展的传输机制，该传输机制允许专用的通信发生在用户计算机上的组件与运行在远程计算机上的组件之间。该协议呈现虚拟的桌面，通常处理图形以及处理诸如 USB、打印机及键盘和鼠标的设备通信。一终端服务器通常主机有多个远程客户机会话。来自终端服务器的图形数据需要编码和加密为图元形式或者在服务器上渲染，生成的位图需要压缩和加密并发送至客户机。

[0112] 编码、渲染和加密运算实质上是非常高计算量的，而且通常要求较高的 CPU 利用率。为了提供好的桌面虚拟化体验，对应的图形保真度应当很高。除代码优化外，可以通过使用额外的硬件资源来降低渲染和加密相关的主机 CPU 周期和网络的占用率，该额外的硬件资源能够执行 RDP 处理过程中所需要的渲染和 / 或压缩任务。此外，可以通过在数据传输前将 CPU 密集型的远程桌面操作卸载至一网络处理器或其他外围设备主机 CPU，来减少主机 CPU 的处理操作。如上所述，该任务卸载操作，也称为烟囱卸载，可以用于卸载 TCP/IP 处理操作，例如校验和、大发送分割 (large send segmentation)、对智能网络结构的 IPSEC 和 TCB 状态处理。因而，利用上述烟囱卸载原理来处理与使用诸如 RDP 的协议来提供远程接入服务相关的一些或全部任务是有益的。

[0113] 图 12 描述了一用于提供远程桌面虚拟化的示范性的体系结构。远程桌面进程被通常栈入传输提供方的顶部，通常为 TCP 的顶部，利用传输层接口 (TLI，transport layer interface) 来发送和接收远程桌面数据包。例如，一个或多个应用程序 1200 可以调用图形和媒体 API，依次产生图形输出，该图形输出进一步由一远程接入处理栈 1210 来处理。网络

传输驱动程序 1220 然后可以处理该数据包的传输过程。

[0114] 图 13 提供了远程桌面处理功能的一进一步示范性的说明。一个或多个应用程序可以在服务器上执行并产生在客户机显示装置上显示的图形。例如应用程序 11310 和应用程序 2 1312 可以产生由图形栈 1318 处理的图形。图形栈 1318 可以包括多个负责呈现图形对象并将其传输至输出设备的图形组件。该组件的例子包括图形设备接口 (GDI)1314 和 DirectX(DX)1316。

[0115] 该应用程序的图形输出被远程接入协议栈 1320 截获，并进一步被处理为命令指令 1322 和位图 1324。命令指令，或绘图指令用于对产生一图形图像或操作一特定的高速缓存所必要的操作进行编码。基本绘图指令一般用于编码绘图操作。每一基本指令可以组织为一组字段，在该字段上应用字段压缩算法。该算法设计为避免发送上一次发送指令后并没有发生改变的字段，并且减小某些字段类型编码后的字段大小，当它们能够通过更小的数据来表示的时候。基本指令的例子包括绘制诸如矩形和线条的图形对象，以及显示文本片段。二级绘图指令可以用于管理高速缓存。

[0116] 位图处理 1324 进一步包括高速缓存 1326 和压缩 1328。该高速缓存过程可以进一步包括一小块化函数 1330 和散列计算函数 1332。远程接入协议可以使用高速缓存来存储绘图图元，例如位图、色表和字符。使用高速缓存技术可以确保在多个绘图操作中使用的项目仅从服务器至客户机发送一次，以此来减小数据通信量。

[0117] 位图处理 1324 的输出和命令指令处理 1322 可以合并为编码指令 1334。远程接入协议可以使用批量压缩 1336 来压缩数据。

[0118] 除了使用数据批量压缩，远程接入协议也可以使用游程编码 (RLE) 规则的变体来实现由服务器发送至客户机的位图数据的压缩。在这里，数据可以由数据成帧 1338，、数据加密 1340 发送，并通过网络 1342 传输。

[0119] 如上所讨论的，例如 TCP 的功能可以使用上面描述的烟囱卸载技术来卸载至一外围设备。如上所述，与诸如 RDP 的协议相关的图形图像的远程化 (remoting) 是通过持续地由服务器向客户机发送更新的位图图片来实现的，该过程是一计算密集型的任务。因此，本发明将该烟囱卸载概念扩展至包括前述的远程接入处理任务。在不同的实施例中，一个或多个与远程桌面虚拟化相关的处理任务可以被卸载至烟囱。在某些实施例中，可以是远程接入处理任务的一个子集被卸载。例如，在一些实施例中，一个或多个位图压缩 1328，高速缓存 1326 以及批量压缩 1336 可以被卸载至一外围设备，例如 NIC，从而使用所公开的卸载原理来处理。在其他实施例中，大多或全部远程接入处理，例如图 13 中描述的任务，可以被卸载至一外围设备，而主机处理器仅保留控制和管理功能。

[0120] 图 14 和图 15 描述了提供对一终端服务器或虚拟机的远程接入的一示范性的操作过程，其中使用一传输层接口来发送和接收远程接入数据单元，该操作过程包括操作 1400，、1402、1404、1406、1408、1410、1412、1414 和 1416。参考图 14，操作 1400 开始该操作过程，操作 1402 为确定外围设备包括用于实现一个或多个指定的远程接入操作任务的任务卸载能力。操作 1404 为发送一外围设备将执行该一个或多个操作任务的指示至外围设备，包括与随后的数据单元共同使用的上下文信息。操作 1406 为使得该一个或多个远程接入操作任务由外围设备执行。

[0121] 操作 1408 示出了，在一实施例中，远程接入是使用远程桌面协议 (RDP，Remote

Desktop Protocol) 来实现的。操作 1410 示出了该操作任务可以提供部分 RDP 处理。或者，操作 1412 示出了该操作任务提供完整的 RDP 处理。操作 1414 示出了一示范性的操作任务包括 RDP 位图压缩。操作 1416 示出了一示范性的操作任务包括 RDP 批量压缩。

[0122] 继续参考图 15，操作 1502 示出了该操作任务包括 RDP 高速缓存。操作 1504 示出了该操作任务根据计算机系统的当前需求来选择性地卸载至外围设备。操作 1504 示出了通过在任务卸载缓冲器中设置至少一个标记指示符启用该任务卸载能力。在一实施例中，每一外围设备驱动程序（例如，参见图 9）可以具有和它相关的一预定义的任务卸载缓冲单元，每一任务卸载缓冲单元包含该设备驱动程序和它对应的外围设备的任务卸载能力。该任务卸载缓冲可以识别外围设备和它的设备驱动程序所支持的特定的任务，也包括与所支持的每一单独的任务相关的任何信息。在一实施例中，一设备驱动程序的任务卸载缓冲的内容可以通过一 NDIS 函数调用来找回。操作 1508 示出了在一实施例中，该外围设备为一网络接口卡 (NIC)。

[0123] 操作 1510 示出了数据包可以传递穿过一分层网络模型。操作 1512 示出了数据包可以包括网络数据和包扩展数据。操作 1514 示出了该数据包指示该外围设备将执行一批操作任务。操作 1516 示出了包扩展数据包括指示至少一个操作任务将由外围设备执行的至少一个数据字段。

[0124] 图 16 描述了一示范性的将远程接入处理任务卸载至一外围设备的操作过程，包括操作 1600、1602 和 1604。参考图 16，操作 1600 开始该操作过程，操作 1602 为向以外围设备发送一数据包，指示该设备将执行以远程接入协议操作任务。操作 1604 为使得上述处理器执行该操作任务。

[0125] 以上提到内容的任何方面可以采用方法、系统、计算机可读的媒体、或任何类型的制造产品来实现。例如，根据图 17，一计算机可读的媒体能在其上存储计算机可执行的用于将远程接入处理任务卸载至一外围设备的指令。该媒体可以包括指令的第一子集 1710，用于确定外围设备包括任务卸载能力；指令的第二子集 1712，用于发送一指示至外围设备，该指示为外围设备将执行上述操作任务之一；以及指令的第三子集 1714，用于使得该远程接入操作任务或更多的远程接入操作任务由外围设备执行。本领域技术人员应当意识到的是，可以使用另外的指令集合来捕捉此处公开的其他不同的内容，而且当前公开的三个指令的子集可以根据当前公开做细节性地变更。

[0126] 例如，该指令可以进一步包括指令 1716，用于使用远程桌面协议 (RDP) 实现上述远程接入。该指令可以进一步包括用于高速缓存和批量压缩的指令 1718，用于多路复用 / 成帧、加密和命令指令编码的指令 1720，以及用于在分层网络模型传递数据包的指令 1722。

[0127] 如上所述，当前公开的主题的各方面可以在经编程的计算机上执行。图 1 以及下面的讨论旨在提供其中可以实现那些方面的合适的计算环境的简要说明。本领域技术人员可领会，图 1 的计算机系统在一些实施例中可实现图 2-4 的服务器和客户机。在这些示例实施例中，服务器和客户机可包括图 1 中所描述的某些或全部组件，且在某些实施例中，服务器和客户机可以各自包括被配置成实例化本发明的特定方面的电路。

[0128] 在本公开中通篇使用的术语“电路”可包括专门硬件组件。在相同实施例或其他实施例中，电路可包括被配置成通过固件或开关执行功能的微处理器。在相同或其他示例

实施例中，电路可包括一个或多个通用处理单元和 / 或多核处理单元等等，当体现可操作以执行功能的逻辑的软件指令被加载到存储器，例如，RAM 和 / 或虚拟存储器中时，这些处理单元可以被配置。在其中电路包括硬件和软件的组合的示例实施例中，实施者可以编写体现逻辑的源代码，且源代码可以被编译为可以由通用处理单元处理的机器可读代码。

[0129] 图 1 描绘了以本发明的各方面来配置的计算系统的示例。计算系统可包括计算机 20 等等，其中包括处理单元 21、系统存储器 22，以及将包括系统存储器在内的各种系统组件耦合到处理单元 21 的系统总线 23。系统总线 23 可以是几种类型的总线结构中的任何一种，包括存储器总线或存储控制器、外围总线、以及使用各种总线体系结构中的任一种的局部总线。系统存储器包括只读存储器 (ROM)24 和随机存取存储器 (RAM)25。基本输入 / 输出系统 26(BIOS) 被存储在 ROM 24 中，该基本输入 / 输出系统 26 包含了诸如在启动期间帮助在计算机 20 内的元件之间传输信息的基本例程。计算机 20 还可以包括用于读写硬盘（未示出）的硬盘驱动器 27、用于读写可移动磁盘 29 的磁盘驱动器 28，以及用于读写诸如 CD ROM 或其他光学介质之类的可移动光盘 31 的光盘驱动器 30。在一些示例实施例中，实施本发明的各方面的计算机可执行指令可存储在 ROM 24、硬盘（未示出）、RAM 25、可移动磁盘 29、光盘 31 和 / 或处理单元 21 的高速缓存中。硬盘驱动器 27、磁盘驱动器 28，以及光驱动器 30 分别通过硬盘驱动器接口 32、磁盘驱动器接口 33，以及光驱动器接口 34 连接到系统总线 23。驱动器以及它们相关联的计算机可读介质为计算机 20 提供了对计算机可读指令、数据结构、程序模块，及其他数据的非易失性存储。虽然此处所描述的环境使用了硬盘、可移动磁盘 29、以及可移动光盘 31，但是，那些本领域普通技术人员应该理解，在操作环境中也可以使用诸如盒式磁带、闪存卡、数字视频盘、伯努利磁带盒、随机存取存储器 (RAM)、只读存储器 (ROM) 等等之类的可以存储可由计算机进行访问的数据的其他类型的计算机可读介质。

[0130] 可以有若干个程序模块存储在硬盘、磁盘 29、光盘 31、ROM 24，和 / 或 RAM25 上，包括操作系统 35、一个或多个应用程序 36、其他程序模块 37、以及程序数据 38。用户可以通过诸如键盘 40 和定点设备 42 之类的输入设备向计算机 20 中输入命令和信息。其他输入设备（未示出）可以包括麦克风、游戏杆、游戏手柄、圆盘式卫星天线、扫描仪等等。这些及其他输入设备常常通过耦合到系统总线的串行端口接口 46 连接到处理单元 21，但是，也可以通过其他接口，如并行端口、游戏端口、通用串行总线 (USB) 端口、来进行连接。显示器 47 或其他类型的显示设备也可以通过诸如视频适配器 48 之类的接口连接到系统总线 23。除了显示器 47 之外，计算机通常还包括其他外围输出设备（未示出），如扬声器和打印机。图 1 的系统也包括主机适配器 55、小型计算机系统接口 (SCSI) 总线 56，以及连接到 SCSI 总线 56 的外部存储装置 62。

[0131] 计算机 20 可以使用到诸如远程计算机 49 之类的一个或多个远程计算机的逻辑连接在联网环境中操作。远程计算机 49 可以是另一计算机、服务器、路由器、网络 PC、对等设备或其他常见的网络节点、虚拟机，并通常包括上文相对于计算机 20 所描述的许多或全部元件，但是在图 1 中只示出了存储器存储设备 50。图 1 中所描绘的逻辑连接可包括局域网 (LAN)51 和广域网 (WAN)52。这样的联网环境在办公室、企业范围的计算机网络、内部网和因特网中是普遍现象。

[0132] 当用于 LAN 网络环境中时，计算机 20 可通过网络接口或适配器 53 连接到 LAN 51。

当用于 WAN 网络环境中时，计算机 20 通常包括调制解调器 54，或用于通过广域网 52（如通过因特网）建立通信的其他装置。调制解调器 54（可以是内置的或外置的）可通过串行端口接口 46 连接到系统总线 23。在联网环境中，相对于计算机 20 所描述的程序模块或其部分可被存储在远程存储器存储设备中。可以理解，所示出的网络连接只是示例，也可以使用用于在计算机之间建立通信链路的其他装置。此外，尽管可以预想当前公开的主题的很多实施例特别适合于计算机系统，但是，本文中没有任何表述旨在将本公开限制于这样的实施例。

[0133] 前述的详细描述通过示例和 / 或操作图阐述了系统和 / 或进程的各实施例。在这样的框图和 / 或示例包含一个或多个功能和 / 或操作的范围内，本领域技术人员将理解，这样的框图，或示例内的每一功能和 / 或操作可以分别地和 / 或共同地通过范围广泛的硬件、软件、固件或几乎其任何组合来实现。

[0134] 虽然已示出和描述了本文中描述的主题内容的特定方面，但是本领域技术人员将明白，基于本文中的教导，可作出改变和修改并且因此所附权利要求的范围旨在涵盖落在本文中描述的主题内容的真实精神和范围内的所有此类改变和修改。

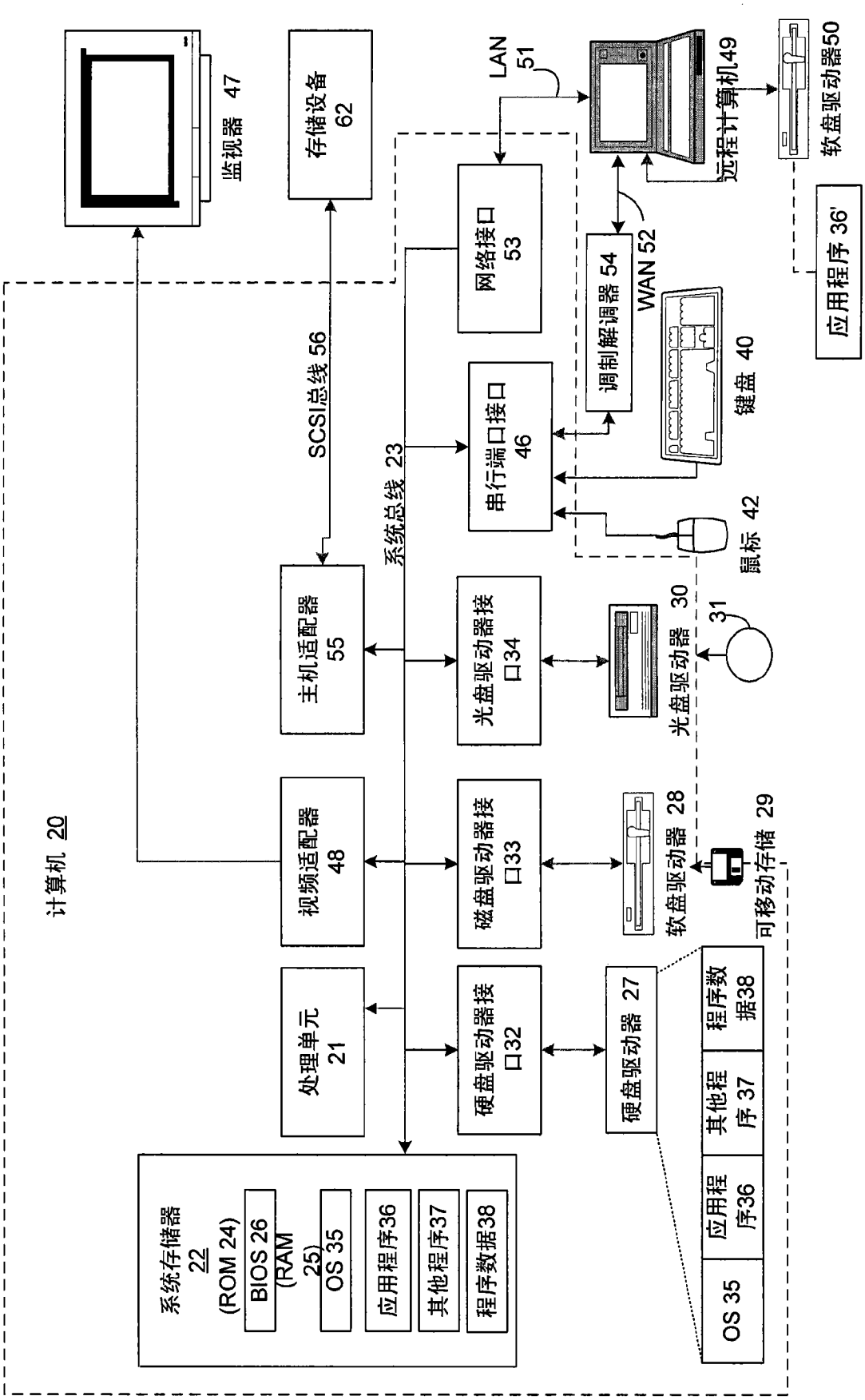

计算机 20
监视器 47
存储设备 62
LAN 51
远程计算机49
软盘驱动器50
应用程序 36'
网络接口 53
调制解调器 54
WAN 52
SCSI总线 56
系统总线 23
串行端口接口 46
键盘 40
鼠标 42
主机适配器 55
光盘驱动器接口34
光盘驱动器 30
视频适配器 48
磁盘驱动器接口33
软盘驱动器 28
可移动存储 29
处理单元 21
硬盘驱动器接口32
硬盘驱动器 27
系统存储器 22 (ROM 24)
BIOS 26
(RAM 25)
OS 35
应用程序36
其他程序37
程序数据38
OS 35
应用程序36
其他程序 37
程序数据38

图 1

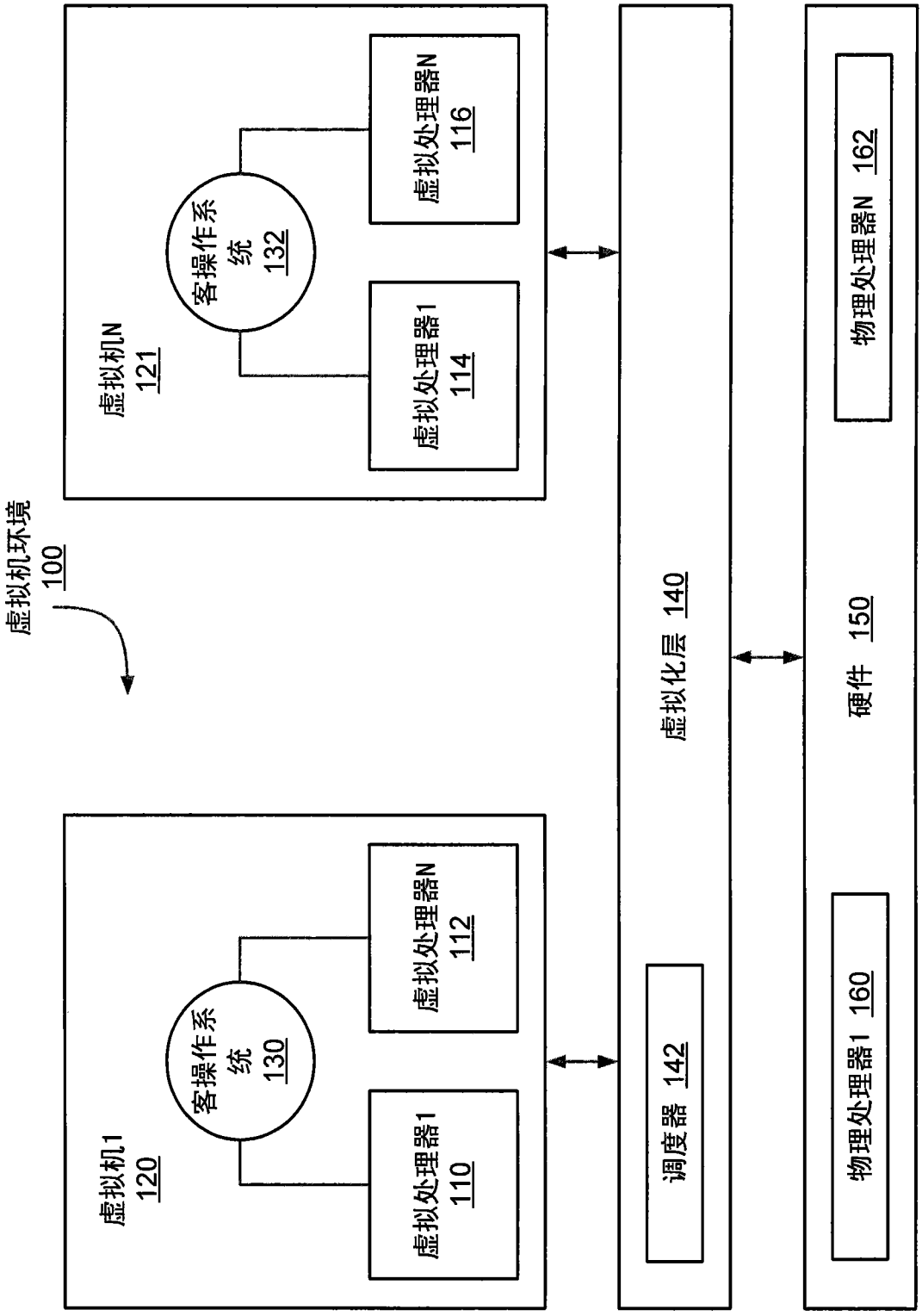

虚拟机环境
100
虚拟机N
121
客操作系统
132
虚拟处理器N
116
虚拟处理器1
114
虚拟机1
120
客操作系统
130
虚拟处理器N
112
虚拟处理器1
110
调度器 142
虚拟化层 140
硬件 150
物理处理器N 162
物理处理器1 160

图 1a

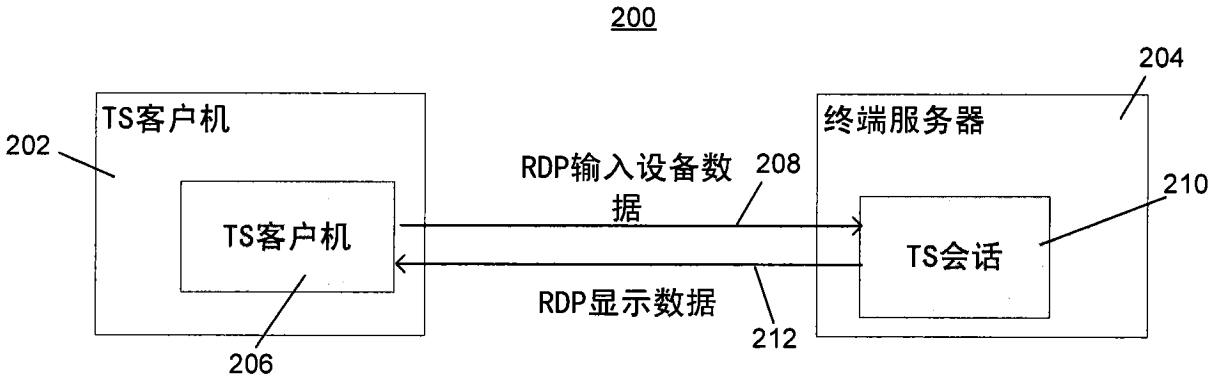

200
TS客户机
202
TS客户机
206
RDP输入设备数据
208
RDP显示数据
212
终端服务器
204
TS会话
210

图 2

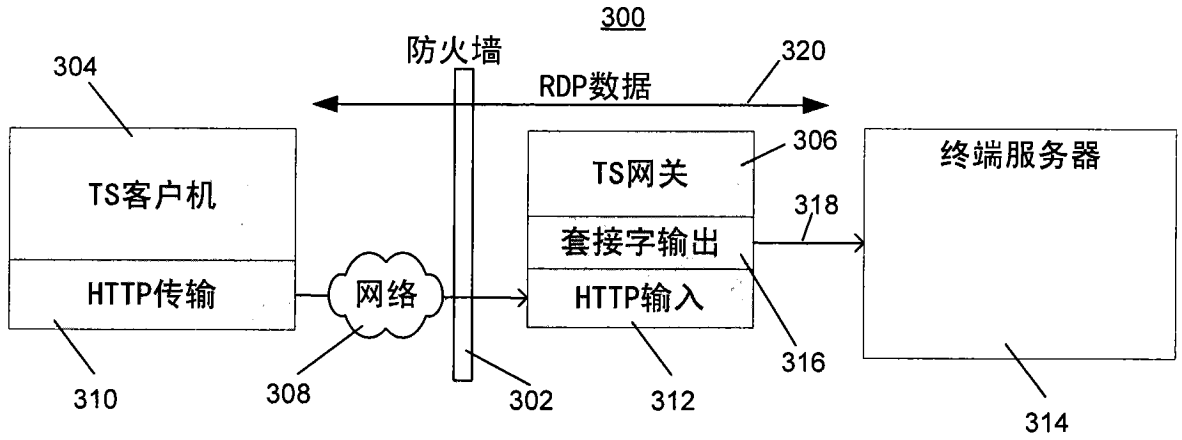

300
防火墙
304
RDP数据
320
TS客户机
HTTP传输
网络
TS网关
套接字输出
HTTP输入
306
318
316
终端服务器
310
308
302
312
314

图 3

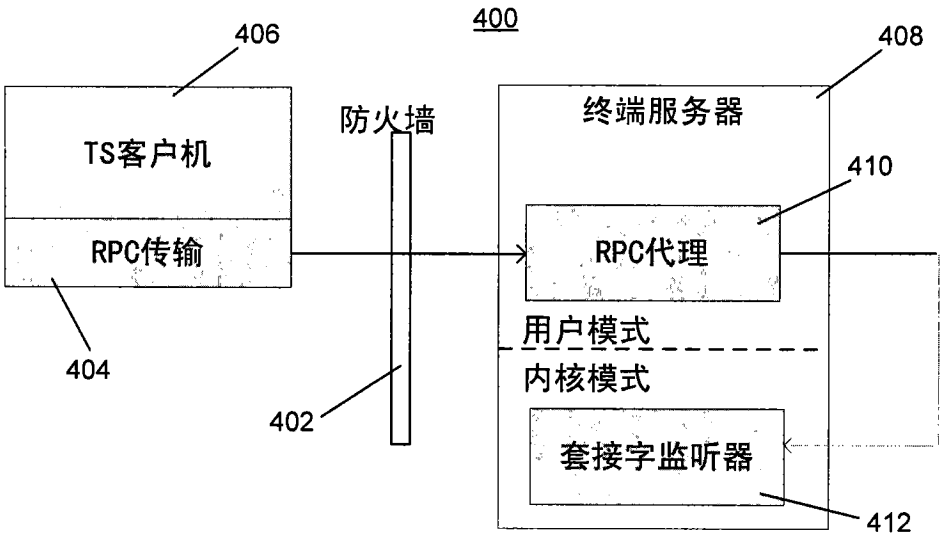

400
406
TS客户机
RPC传输
404
防火墙
402
408
终端服务器
410
RPC代理
用户模式
内核模式
套接字监听器
412

图 4

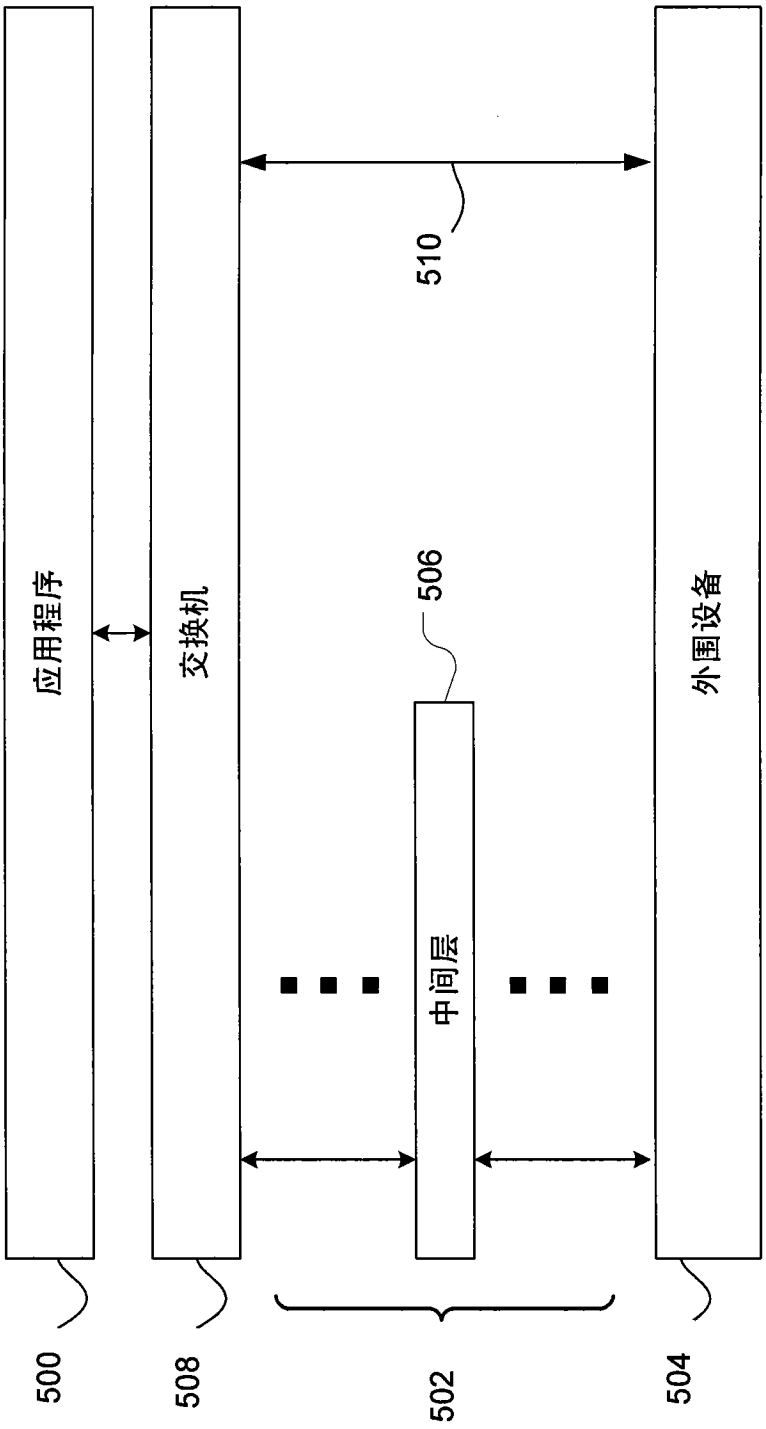


图 5

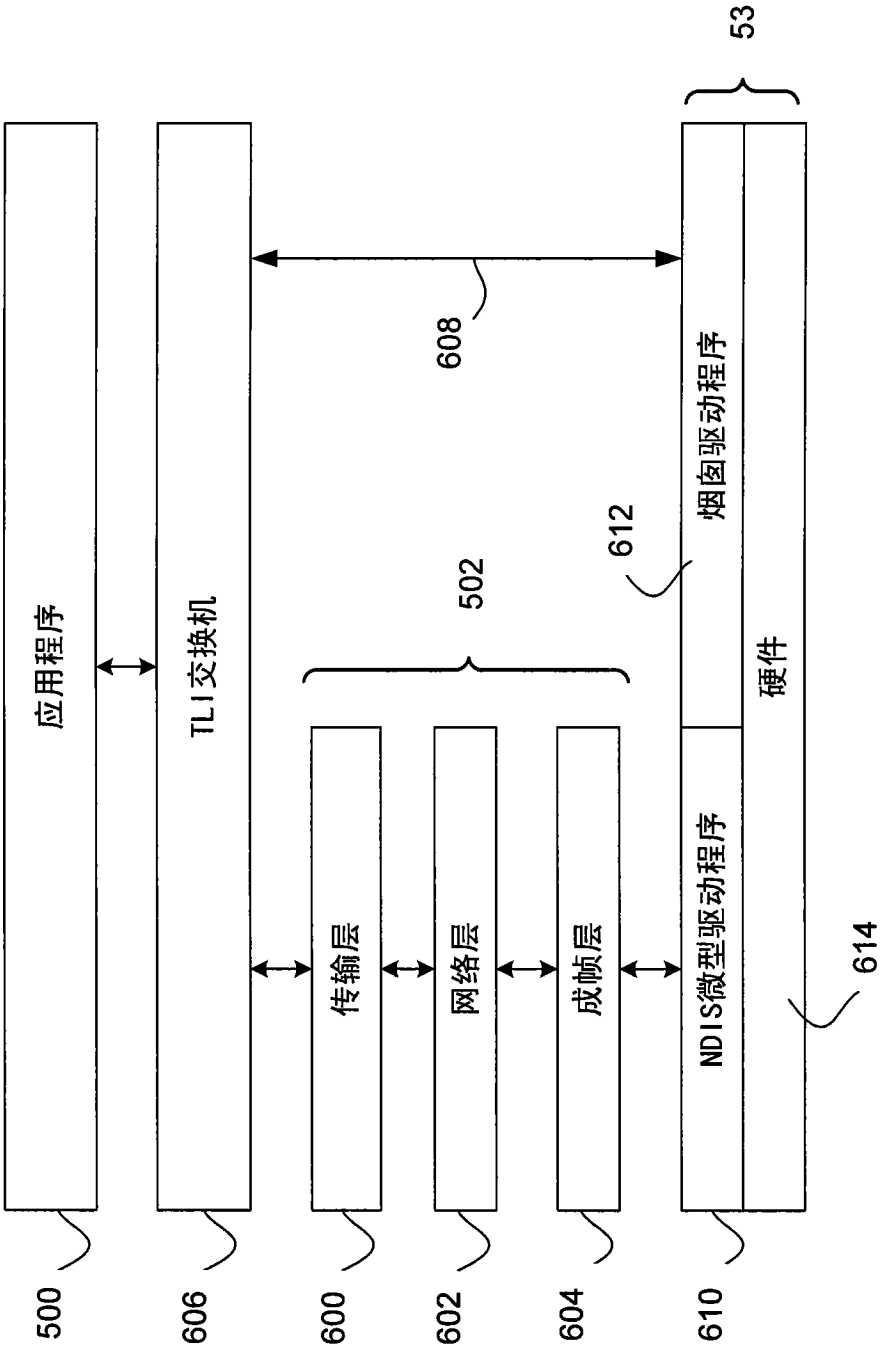

53
608
612
502
应用程序
TLI交换机
传输层
网络层
成帧层
烟囱驱动程序
NDIS微型驱动程序
硬件
614
500
606
600
602
604
610

图 6

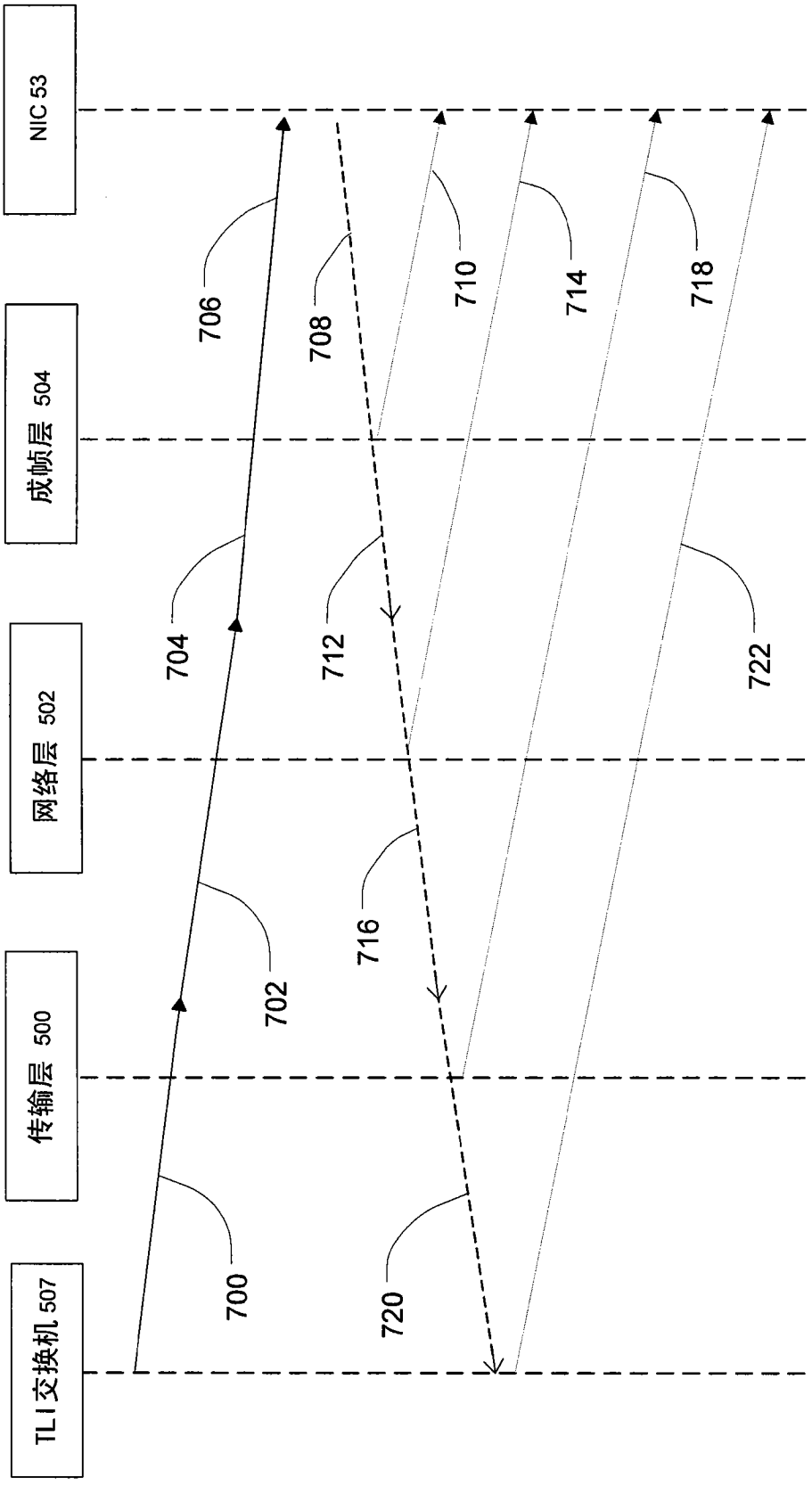

NIC 53
成帧层 504
网络层 502
传输层 500
TLI交换机 507
700
702
704
706
708
710
712
714
716
718
720
722

图 7

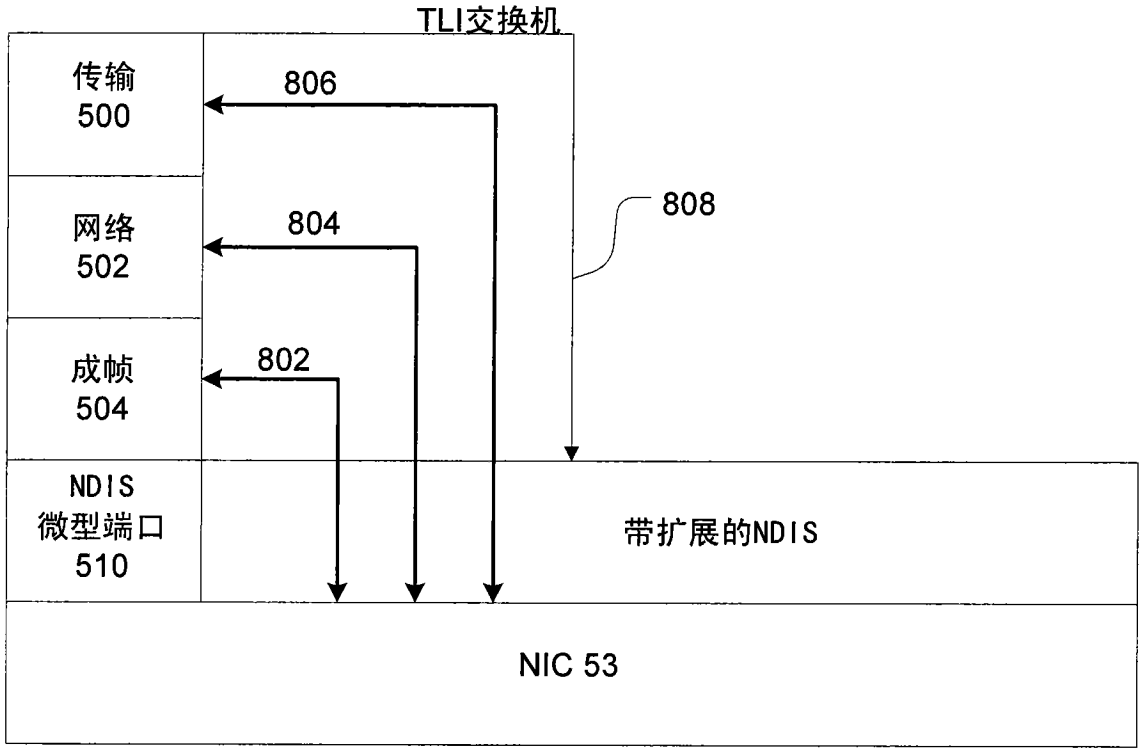

TLI交换机
传输
500
网络
502
成帧
504
NDIS
微型端口
510
806
804
802
808
带扩展的NDIS
NIC 53

图 8

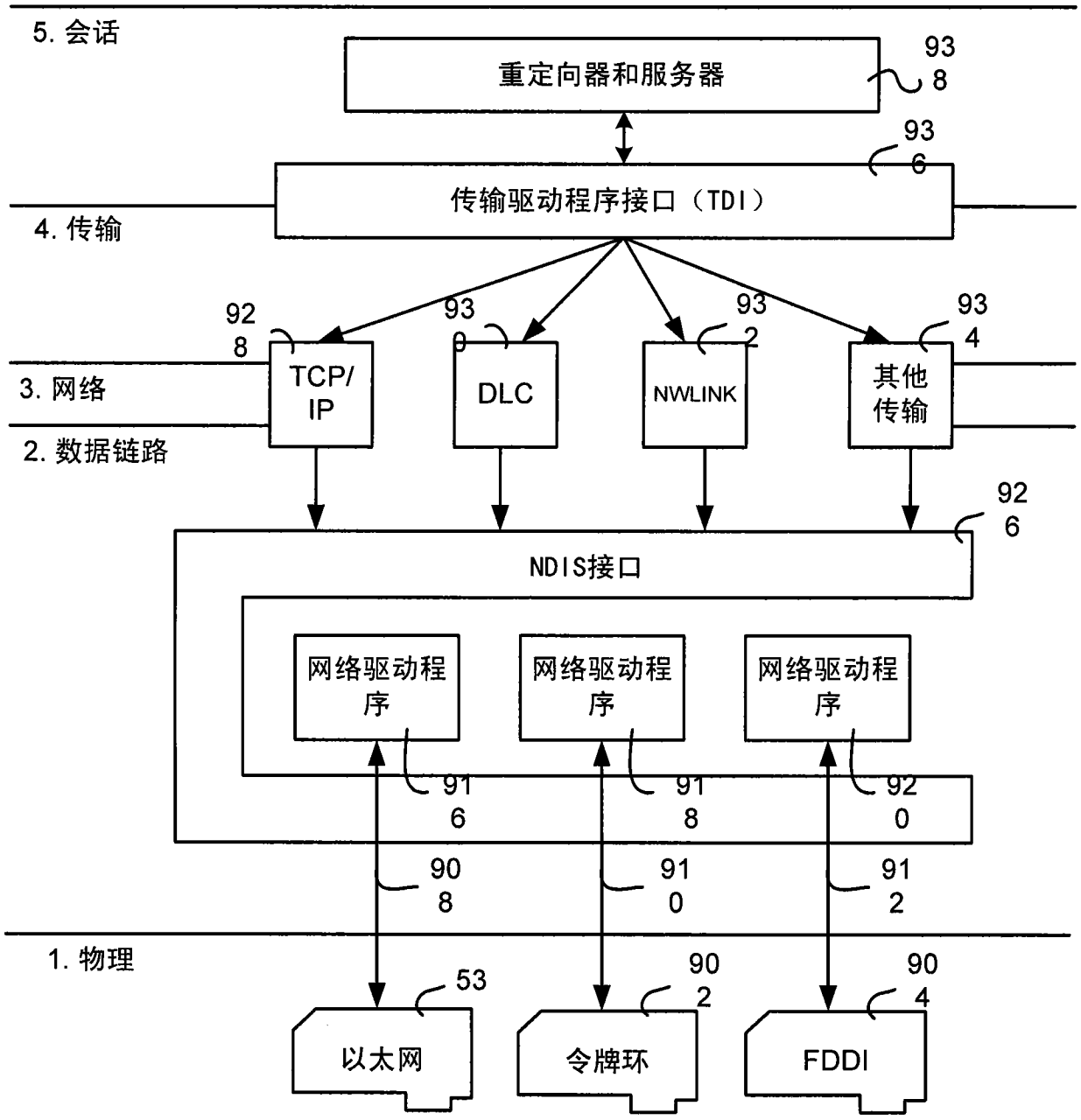

5. 会话
重定向器和服务器
938
936
传输驱动程序接口（TDI）
4. 传输
928
TCP/IP
930
DLC
932
NWLINK
934
其他传输
3. 网络
2. 数据链路
926
NDIS接口
网络驱动程序
网络驱动程序
网络驱动程序
916
918
920
908
910
912
1. 物理
53
以太网
902
令牌环
904
FDDI

图 9

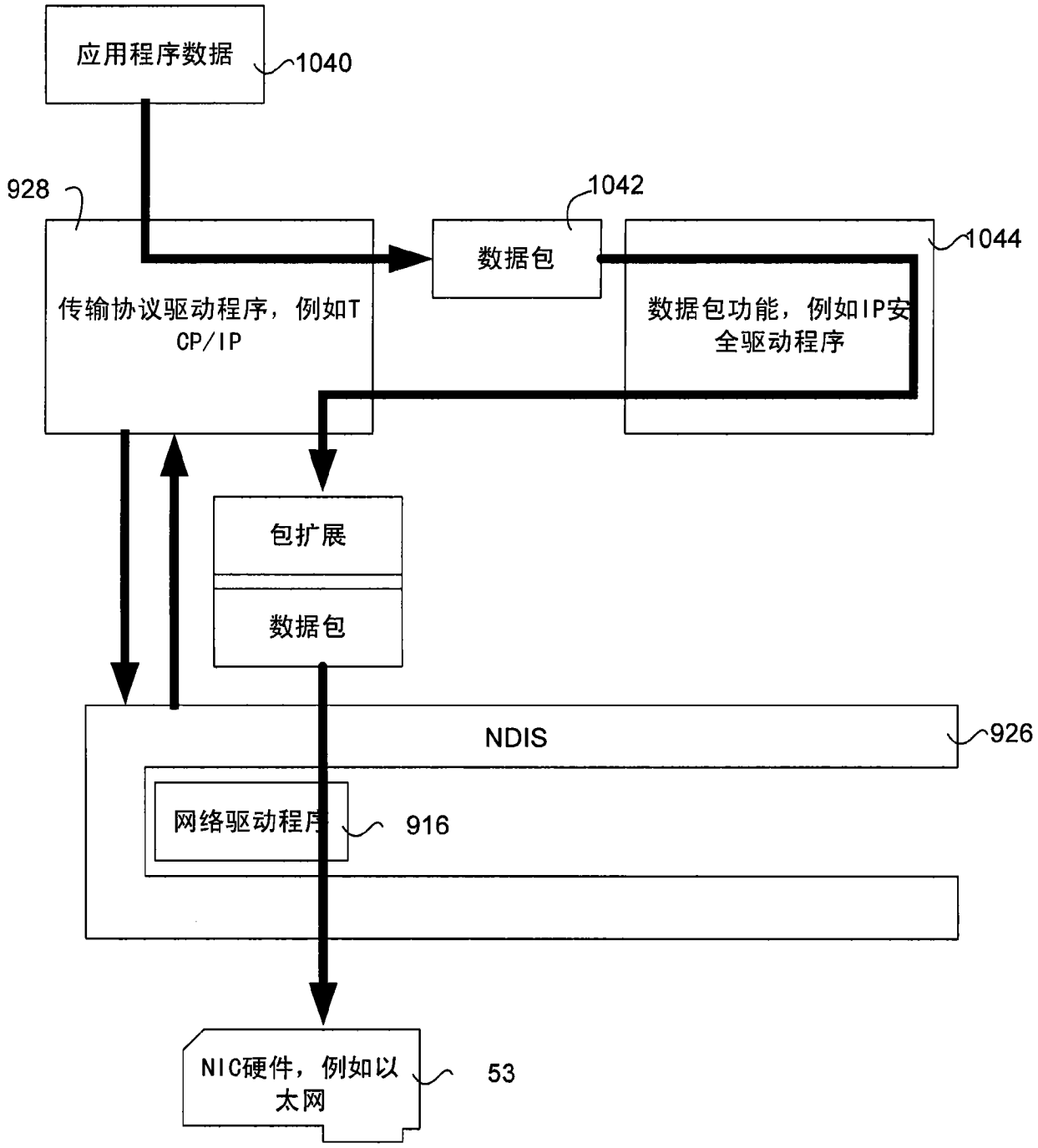

应用程序数据
1040
928
1042
1044
数据包
传输协议驱动程序，例如TCP/IP
数据包功能，例如IP安全驱动程序
包扩展
数据包
NDIS
926
网络驱动程序
916
NIC硬件，例如以太网
53

图 10

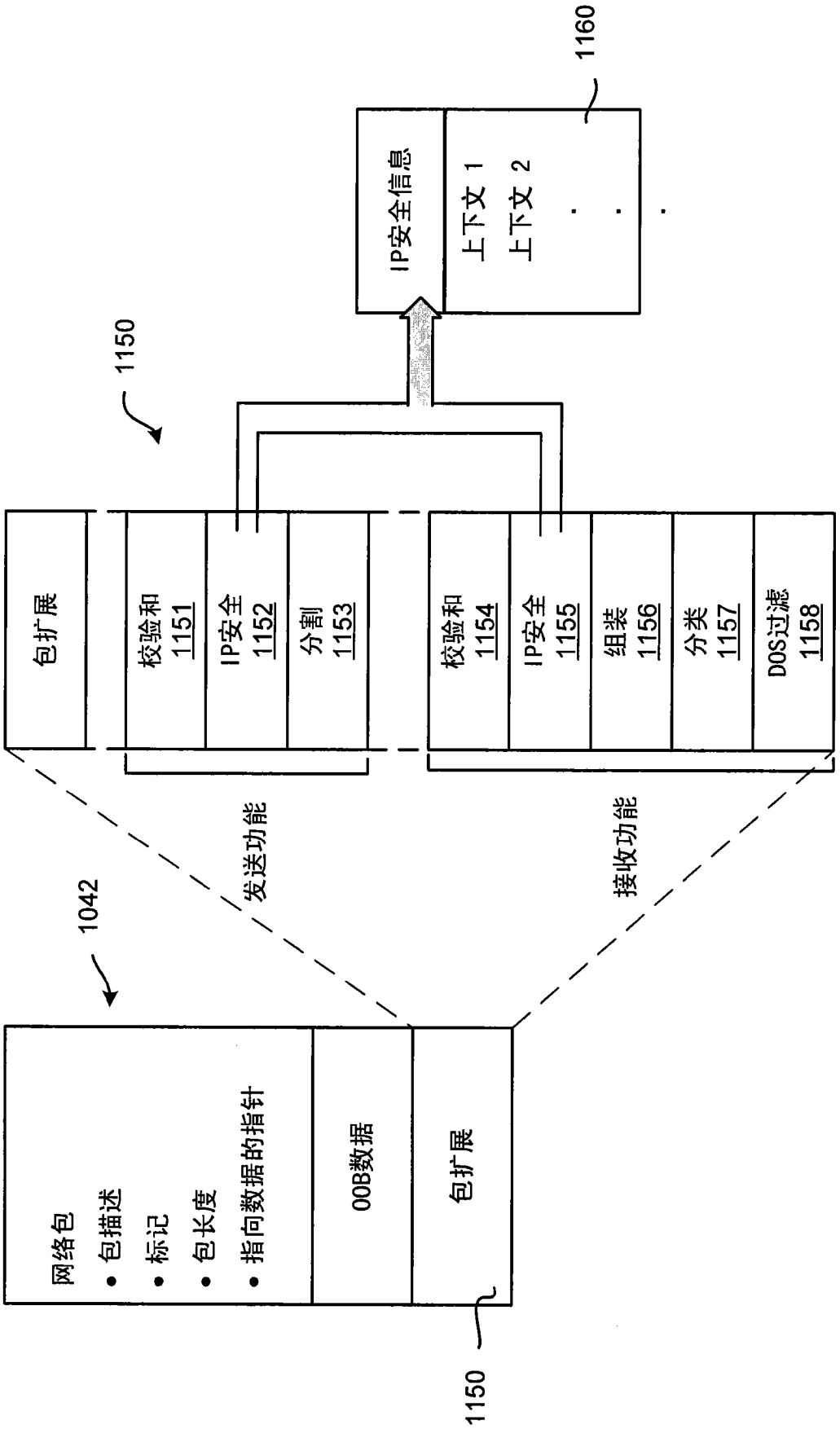

1160
IP安全信息
上下文 1
上下文 2
1150
包扩展
校验和 1151
IP安全 1152
分割 1153
校验和 1154
IP安全 1155
组装 1156
分类 1157
DOS过滤 1158
发送功能
接收功能
1042
网络包
• 包描述
• 标记
• 包长度
• 指向数据的指针
OOB数据
包扩展
1150

图 11

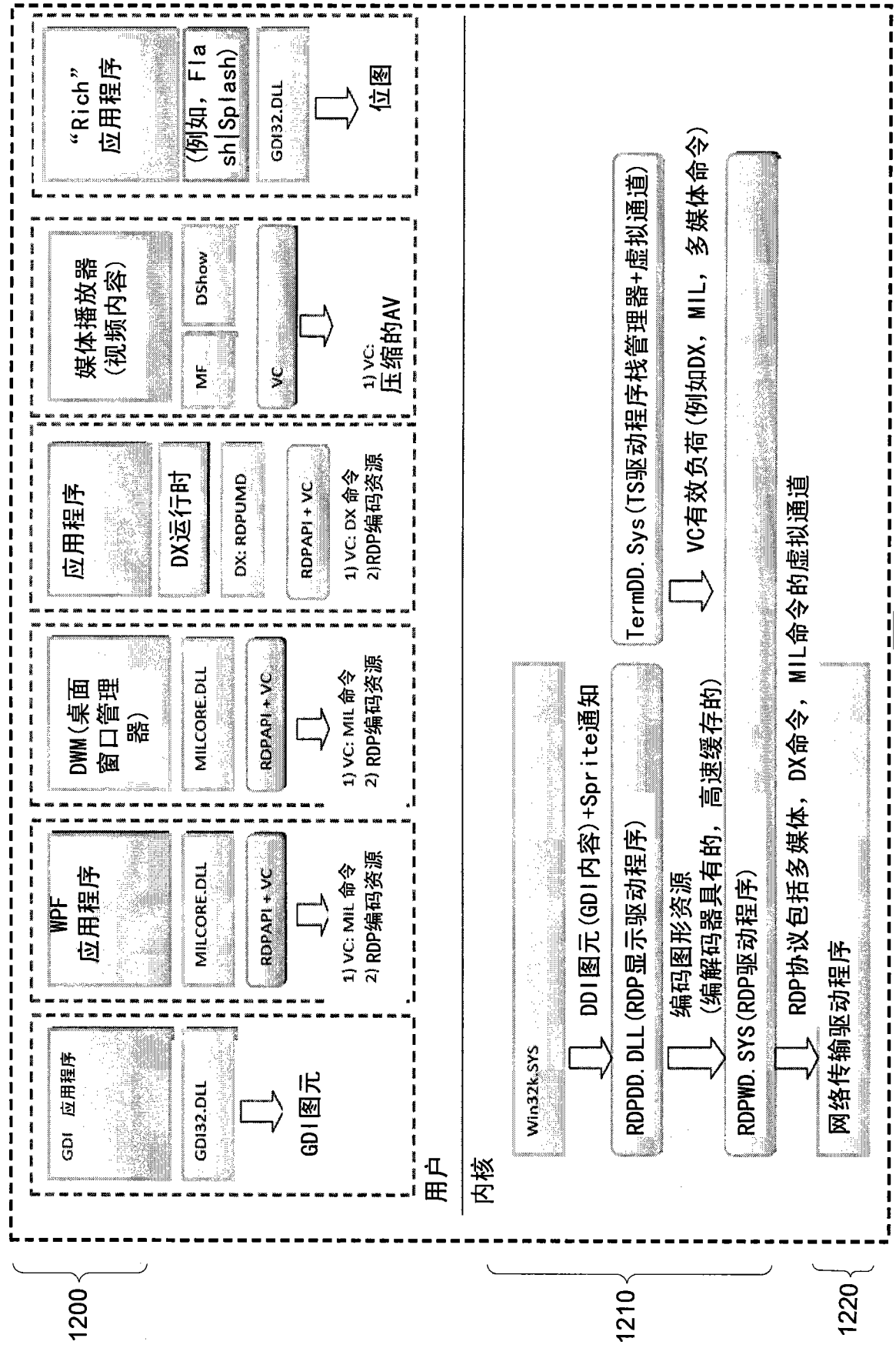

1200
1210
1220
用户
内核
GDI 应用程序
GDI32.DLL
GDI图元
WPF
应用程序
MILCORE.DLL
RDPAPI + VC
1) VC: MIL 命令
2) RDP编码资源
DWM(桌面
窗口管理
器)
MILCORE.DLL
RDPAPI + VC
1) VC: MIL 命令
2) RDP编码资源
应用程序
DX运行时
DX: RDPUMD
RDPAPI + VC
1) VC: DX 命令
2)RDP编码资源
媒体播放器
(视频内容)
MF
DShow
VC
1) VC:
压缩的AV
"Rich"
应用程序
(例如，Flash|Splash)
GDI32.DLL
位图
Win32k.SYS
DDI图元(GDI内容)+Sprite通知
RDPDD.DLL(RDP显示驱动程序)
编码图形资源
(编解码器具有的，高速缓存的)
RDPWD.SYS(RDP驱动程序)
RDP协议包括多媒体，DX命令，MIL命令的虚拟通道
网络传输驱动程序
TermDD.Sys(TS驱动程序栈管理器+虚拟通道)
VC有效负荷(例如DX，MIL，多媒体命令)

图 12

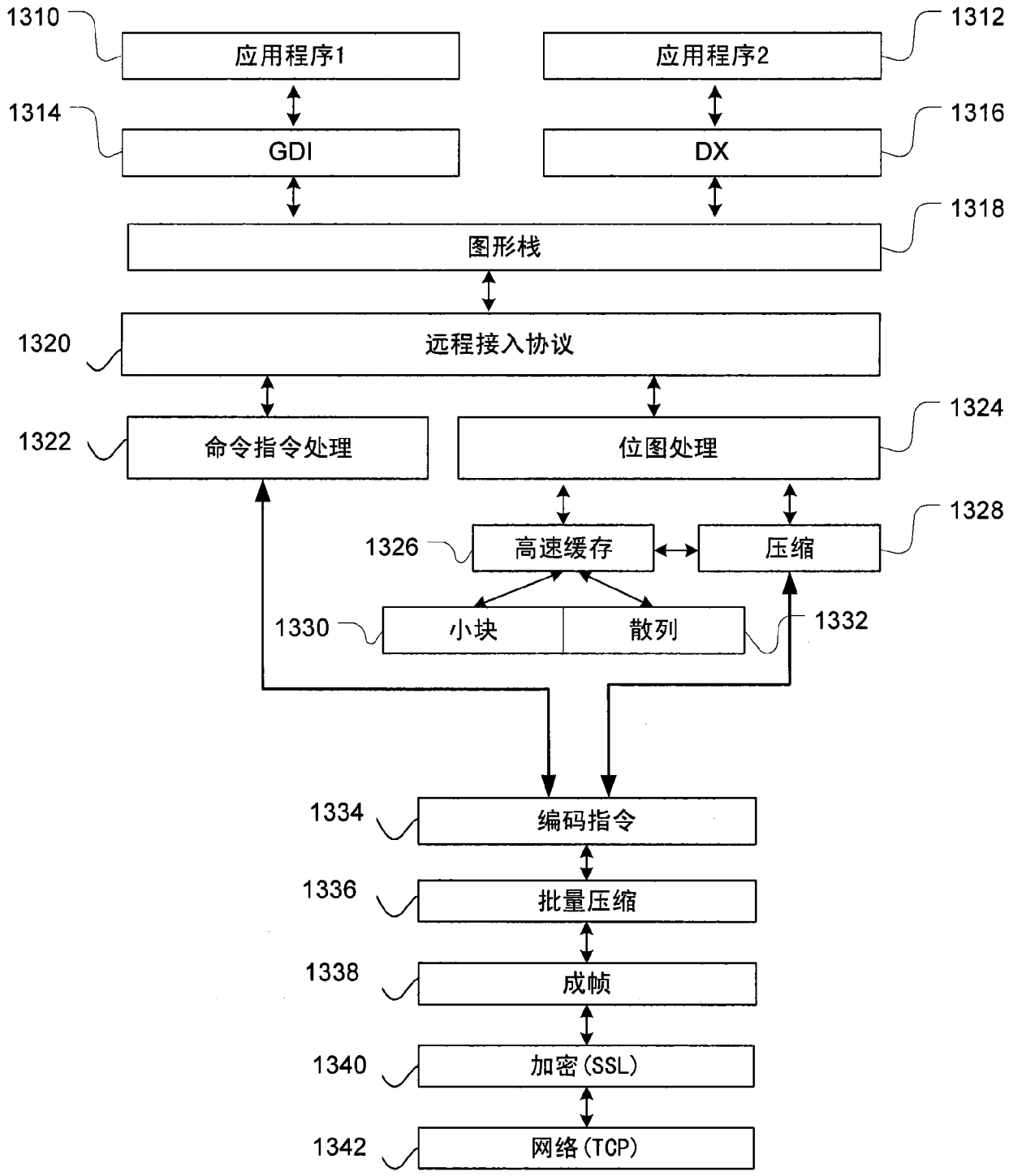

1310
应用程序1
1312
应用程序2
1314
GDI
1316
DX
1318
图形栈
1320
远程接入协议
1322
命令指令处理
位图处理
1324
1326
高速缓存
压缩
1328
1330
小块
散列
1332
1334
编码指令
1336
批量压缩
1338
成帧
1340
加密(SSL)
1342
网络(TCP)

图 13

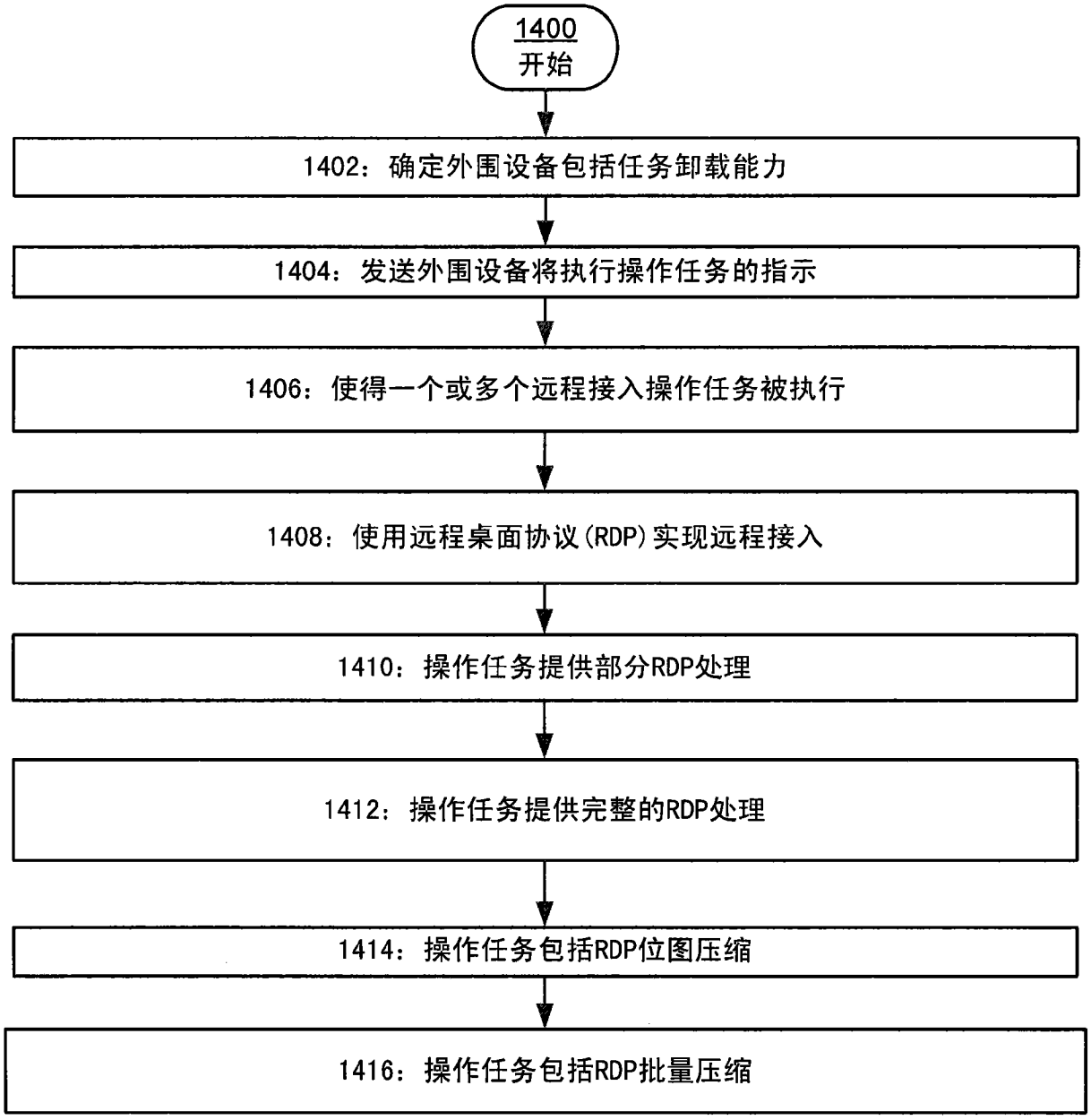

1400
开始
1402：确定外围设备包括任务卸载能力
1404：发送外围设备将执行操作任务的指示
1406：使得一个或多个远程接入操作任务被执行
1408：使用远程桌面协议(RDP)实现远程接入
1410：操作任务提供部分RDP处理
1412：操作任务提供完整的RDP处理
1414：操作任务包括RDP位图压缩
1416：操作任务包括RDP批量压缩

图 14

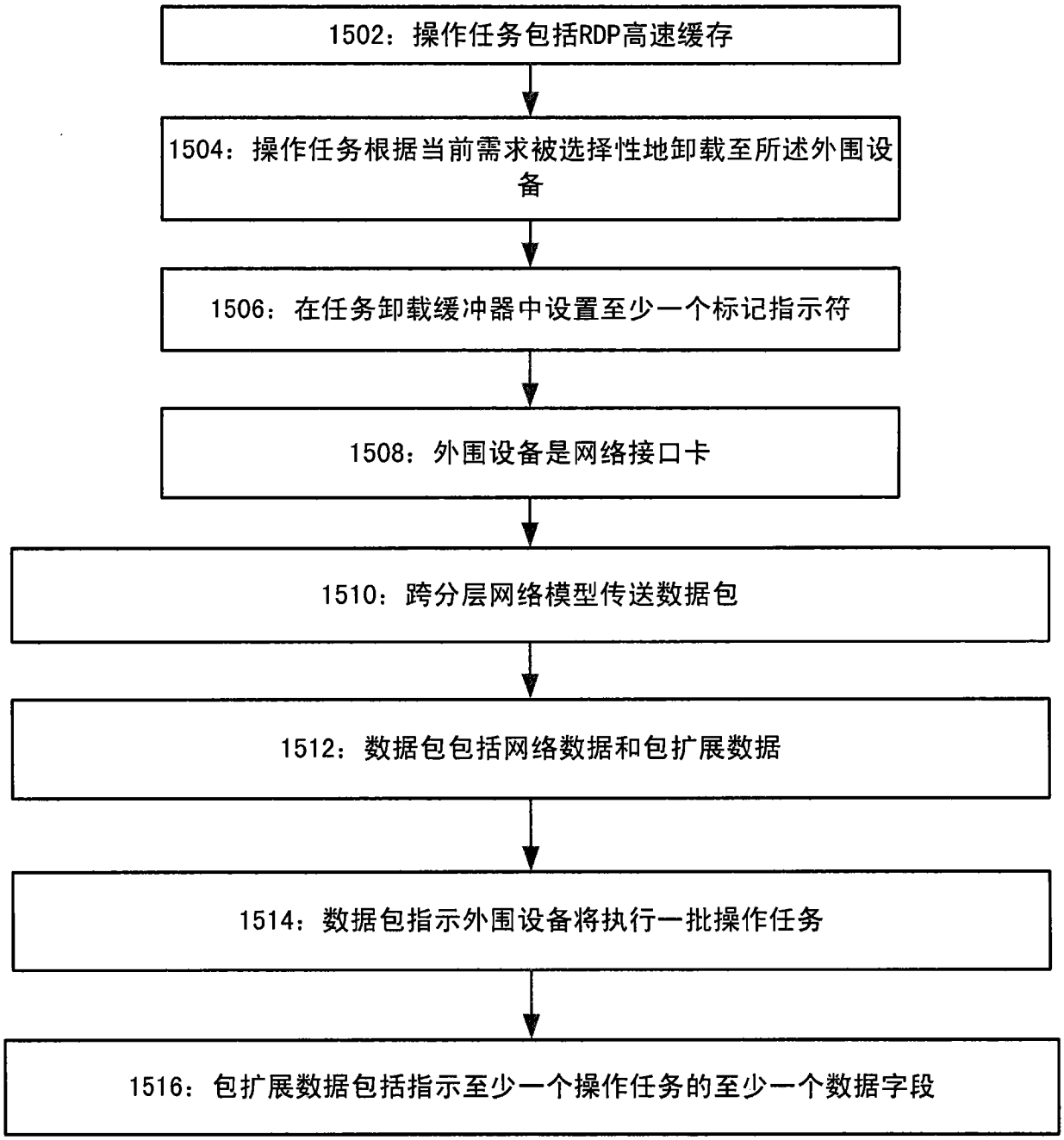

1502：操作任务包括RDP高速缓存
1504：操作任务根据当前需求被选择性地卸载至所述外围设备
1506：在任务卸载缓冲器中设置至少一个标记指示符
1508：外围设备是网络接口卡
1510：跨分层网络模型传送数据包
1512：数据包包括网络数据和包扩展数据
1514：数据包指示外围设备将执行一批操作任务
1516：包扩展数据包括指示至少一个操作任务的至少一个数据字段

图 15

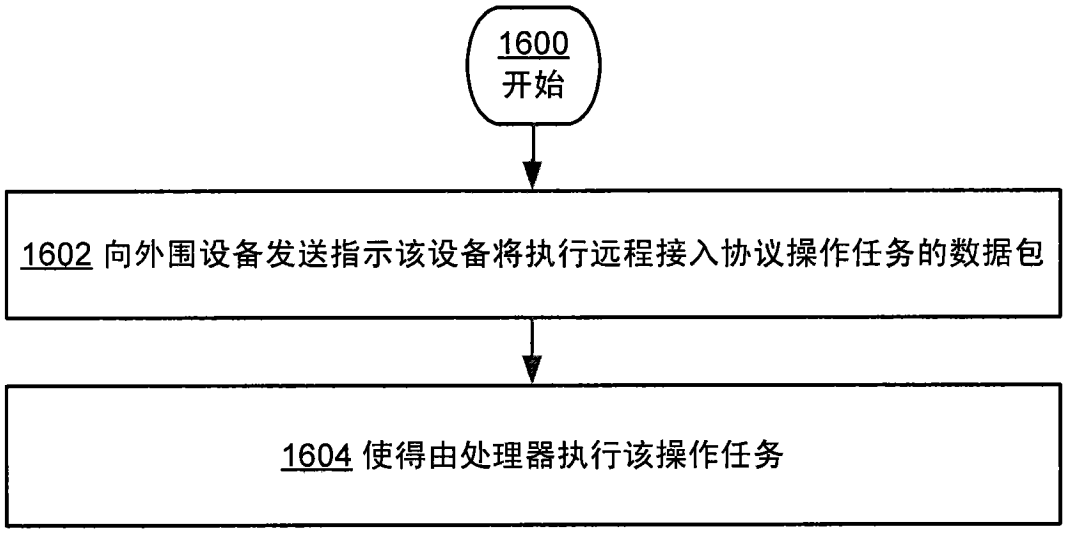

1600
开始
1602 向外围设备发送指示该设备将执行远程接入协议操作任务的数据包
1604 使得由处理器执行该操作任务

图 16

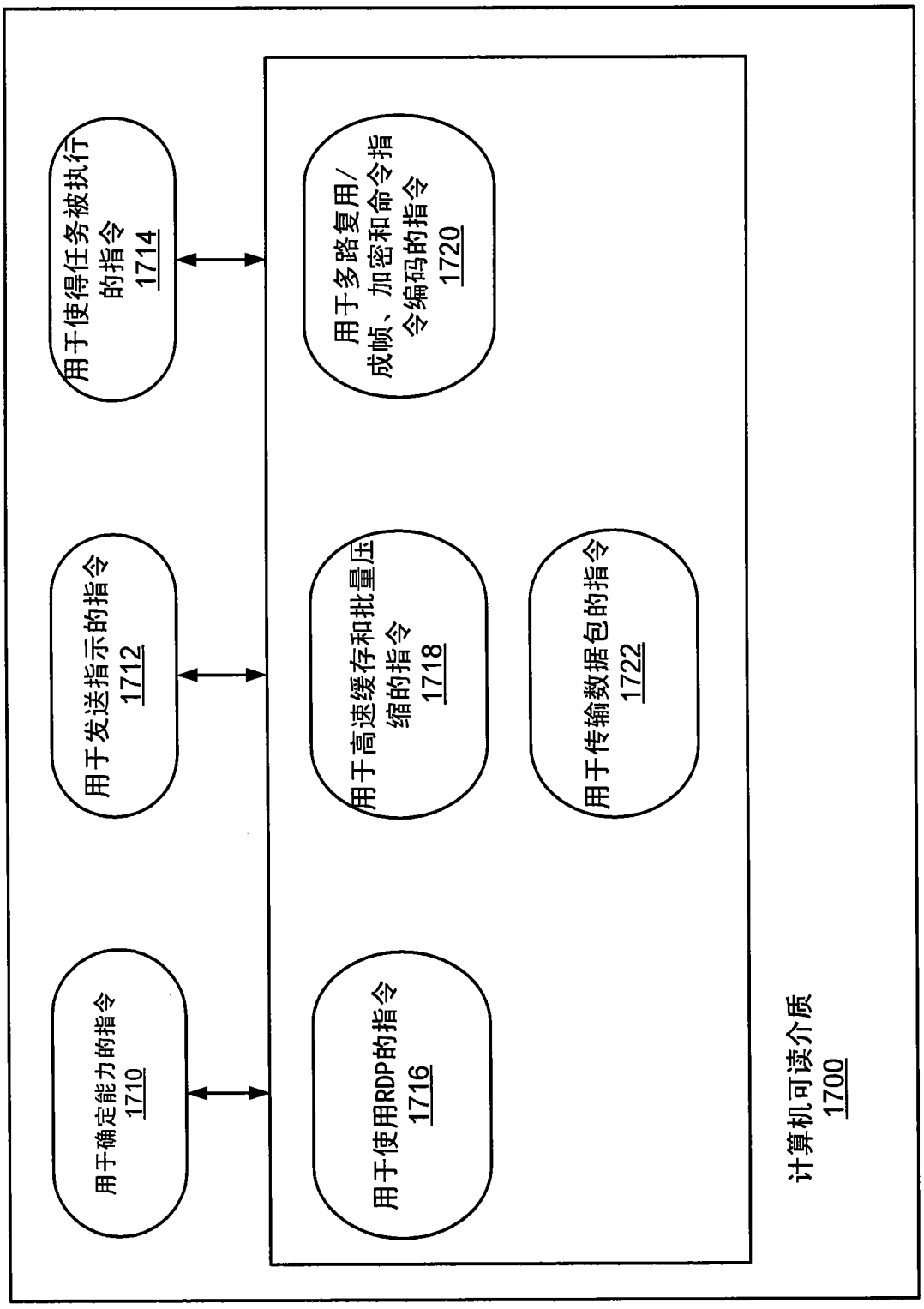

用于确定能力的指令
1710
用于发送指示的指令
1712
用于使得任务被执行的指令
1714
用于使用RDP的指令
1716
用于高速缓存和批量压缩的指令
1718
用于传输数据包的指令
1722
用于多路复用/成帧、加密和命令指令编码的指令
1720
计算机可读介质
1700

图 17