[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
H04L 12/56
H04L 12/24 H04L 29/06

[12] 发明专利申请公开说明书

[21] 申请号 200410048864.7

[43] 公开日 2005 年 2 月 2 日

[11] 公开号 CN 1574792A

[22] 申请日 2004.6.3
[21] 申请号 200410048864.7
[30] 优先权
[32] 2003. 6. 6 [33] US [31] 10/456，770
[71] 申请人 微软公司
地址 美国华盛顿州
[72] 发明人 B·D·斯旺达 G·S·帕尔
N·S·S·N·劳

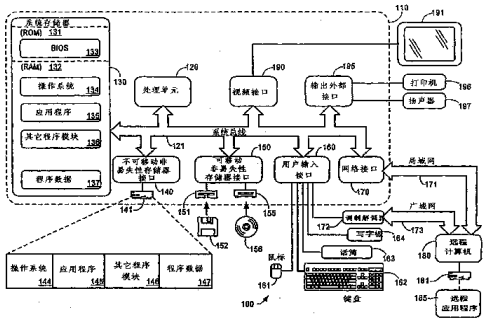
[74] 专利代理机构 上海专利商标事务所
代理人 李家麟

权利要求书 5 页 说明书 34 页 附图 12 页

[54] 发明名称 用于执行网络防火墙的基于多层的方法

[57] 摘要

提供了一种方法，用于在防火墙结构中执行防火墙。 该防火墙结构包括多个网络层和第一个防火墙引擎。 这些层将信息包和信息包信息发送到这第一个防火墙引擎，保持信息包上下文并将其传递到随后的各个层，并且处理这些信息包。 这第一个防火墙引擎将该信息包信息与一个或多个被安装的滤波器进行比较，并将动作返回到这些层，从而指出如何处置该信息包。



ISSN1008-4274

1. 一种用于在请求阶段执行防火墙策略的方法，该请求阶段是防火墙构架中的多个阶段之一，该防火墙构架进一步包括具有多个被安装的滤波器的防火墙引擎，其特征在于，包括：

通过该请求阶段，从这多个阶段中的前一个阶段接收信息包；

通过该请求阶段，识别与该信息包关联的一组参数；

发出分类调用，该分类调用包括与该信息包关联的这个参数集；

响应于该分类调用，根据这多个被安装的滤波器中的至少一个滤波器所指定的该防火墙策略来接收动作。

2. 如权利要求 1 所述的方法，其特征在于，该动作是指令，用于允许该信息包继续进行网络遍历，进一步包括：

根据层协议来处理该信息包；以及，

将该信息包发送到这多个阶段中的下一个阶段。

3. 如权利要求 1 所述的方法，其特征在于，该信息包是被指定到网络设备去的出站信息包；并且，其中，这个参数集包括根据该请求阶段所执行的协议而被加入该信息包的信息。

4. 如权利要求 1 所述的方法，其特征在于，该信息包是从网络设备接收的入站信息包；并且，其中，这个参数集包括信息，该请求阶段根据该请求阶段所执行的协议来从该入站信息包中分析该信息。

5. 如权利要求 1 所述的方法，其特征在于，这多个阶段在操作系统的核心模式内执行。

6. 如权利要求 1 所述的方法，其特征在于，这多个阶段在操作系统的用户模式内执行。

7. 如权利要求 1 所述的方法，其特征在于，进一步包括：

从这前一个阶段接收信息包上下文数据结构，该信息包上下文包括与该信息包关联的先前的阶段数据；以及，

通过增加这个参数集，来修改该信息包上下文数据结构。

8. 如权利要求 7 所述的方法，其特征在于，这个参数集中的参数包括该请求阶段的标识、该参数类型和值。

9. 如权利要求 7 所述的方法，其特征在于，该请求阶段将该修改过的信息包上下文数据结构发送到这多个阶段中的下一个阶段。

10. 一种用于在包括一组被安装的滤波器的防火墙引擎中执行防火墙策略的方法，这些被安装的滤波器每个都包括一组滤波器条件和一个关联的动作，其特征在于，包括：

接收一组信息包参数，这些信息包参数包括与请求层关联的第一信息包信息以及与信息包上下文数据结构关联的第二信息包信息；

识别一组匹配滤波器，这个匹配滤波器集中的每个滤波器具有对应于这些信息包参数的滤波器条件；以及，

从这些匹配滤波器中的至少一个滤波器中识别该关联的动作。

11. 如权利要求 10 所述的方法，其特征在于，这个匹配滤波器集中的每个滤波器具有优先级，并且，来自最高优先级滤波器的该关联的动作是非终止动作，进一步包括：

从这个匹配滤波器集中的一个或多个较低优先级滤波器中识别该关联的动作，直到达到终止动作为止。

12. 如权利要求 10 所述的方法，其特征在于，来自这个匹配滤波器集的滤波器识别呼出(callout)模块，进一步包括：

将这些信息包参数和来自这个匹配滤波器集的该滤波器的标识发送到该呼出模块。

13. 如权利要求 12 所述的方法，其特征在于，该呼出修改该信息包上下

文。

14. 如权利要求 10 所述的方法，其特征在于，该防火墙引擎在操作系统的用户模式中执行。

15. 如权利要求 10 所述的方法，其特征在于，该防火墙引擎在操作系统的核心模式中执行。

16. 一种用于允许在启动网络设备与响应网络设备之间进行网络通信的方法，该响应网络设备包括用于防止未经请求的网络通信的防火墙，其特征在于，包括：

根据密钥商议协议，来创建允许入站信息包的第一个防火墙滤波器；

根据该密钥商议协议，在该启动网络设备与该响应网络设备之间实施成功的密钥商议；

验证该启动设备的身份，作为该密钥商议协议的一部分；以及，

创建第二个防火墙滤波器，该防火墙滤波器允许从该启动网络设备那里被发送的入站信息包。

17. 如权利要求 16 所述的方法，其特征在于，这第二个防火墙滤波器允许从该启动网络设备那里被发送的、符合安全协议的信息包。

18. 如权利要求 16 所述的方法，其特征在于，这第二个防火墙滤波器允许指定该启动设备的地址的信息包。

19. 如权利要求 16 所述的方法，其特征在于，该密钥商议是 IKE 商议。

20. 一种用于执行计算机可读指令的计算机可读介质，这些计算机可读指令用于在请求阶段执行防火墙策略，该请求阶段是防火墙构架中的多个阶段之一，该防火墙构架进一步包括具有多个被安装的滤波器的防火墙引擎，其特征在于，包括：

通过该请求阶段，从这多个阶段中的前一个阶段接收信息包；

通过该请求阶段，识别与该信息包关联的一组参数；

发出分类调用，该分类调用包括与该信息包关联的这个参数集；

响应于该分类调用，根据这多个被安装的滤波器中的至少一个滤波器所指定的该防火墙策略来接收动作。

21. 如权利要求 20 所述的计算机可读介质，其特征在于，该动作是指令，用于允许该信息包继续进行网络遍历，进一步包括：

根据层协议来处理该信息包；以及，

将该信息包发送到这多个阶段中的下一个阶段。

22. 如权利要求 20 所述的计算机可读介质，其特征在于，进一步包括：

从这前一个阶段接收信息包上下文数据结构，该信息包上下文包括与该信息包关联的先前的阶段数据；以及，

通过增加这个参数集，来修改该信息包上下文数据结构。

23. 一种用于执行计算机可读指令的计算机可读介质，这些计算机可读指令用于在包括一组被安装的滤波器的防火墙引擎中执行防火墙策略，这些被安装的滤波器每个都包括一组滤波器条件和一个关联的动作，其特征在于，包括：

接收一组信息包参数，这些信息包参数包括与请求层关联的第一信息包信息以及与信息包上下文数据结构关联的第二信息包信息；

识别一组匹配滤波器，这个匹配滤波器集中的每个滤波器具有对应于这些信息包参数的滤波器条件；以及，

从这些匹配滤波器中的至少一个滤波器中识别该关联的动作。

24. 如权利要求 23 所述的计算机可读介质，其特征在于，这个匹配滤波器集中的每个滤波器具有优先级，并且，来自最高优先级滤波器的该关联的动作是非终止动作，进一步包括：

从这个匹配滤波器集中的一个或多个较低优先级滤波器中识别该关联的动作，直到达到终止动作为止。

25. 如权利要求23所述的计算机可读介质，其特征在于，来自这个匹配滤波器集的滤波器识别呼出模块，进一步包括：

将这些信息包参数和来自这个匹配滤波器集的该滤波器的标识发送到该呼出模块。

26. 一种用于执行计算机可读指令的计算机可读介质，这些计算机可读指令用于允许在启动网络设备与响应网络设备之间进行网络通信，该响应网络设备包括用于防止未经请求的网络通信的防火墙，其特征在于，包括：

根据密钥商议协议，来创建允许入站信息包的第一个防火墙滤波器；

根据该密钥商议协议，在该启动网络设备与该响应网络设备之间实施成功的密钥商议；

验证该启动设备的身份，作为该密钥商议协议的一部分；以及，

创建第二个防火墙滤波器，该防火墙滤波器允许从该启动网络设备那里被发送的入站信息包。

27. 如权利要求26所述的计算机可读介质，其特征在于，这第二个防火墙滤波器允许从该启动网络设备那里被发送的、符合安全协议的信息包。

28. 如权利要求26所述的计算机可读介质，其特征在于，这第二个防火墙滤波器允许指定该启动设备的地址的信息包。

用于执行网络防火墙的基于多层的方法

相关申请

本申请所包含的主题与跟本申请同日提交的专利申请《多层防火墙结构》(代理编号：221038)、《用于结合多项网络策略的方法和构架》(代理编号：221041)和《用于管理基于网络滤波器的策略的方法》(代理编号：221037)的主题有关，这些专利申请的揭示说明被特别包括于此，用作参考。

技术领域

本发明一般涉及计算机系统和网络安全性。更具体地说，本发明涉及一种在网络设备中执行防火墙的方法。

背景技术

网络协议的设计用于促进通过公开的数据交换而在各个网络设备之间进行通信。这种公开的数据交换大大增强了网络设备完成任务的用途，同时，它也产生一些问题，这是因为网络协议不是为网络安全性而设计，因此一般不提供网络安全性。被耦合到公用网和专用网(例如，局域网(LANs)、广域网(WANs)、内联网和因特网)的计算机容易受到直接或间接地与该网络耦合的其他网络设备的恶意攻击。这类恶意的攻击包括盗窃数据、“服务拒绝”(DOS)攻击、计算机病毒扩散和类似的攻击。当将计算机耦合到网络时，会出现其他有关的事项(例如，控制儿童对不合需要的或不适当的web站点的访问)。

防火墙一般是用来保护用户个人、网络设备和网络避免遭遇恶意攻击的一种工具，同时，也增加了通过实施策略来控制网络上的数据交换的能力。通过检查网络信息包，并通过根据该检查来确定是应该允许还是相反地阻滞这些信息包进一步穿过网络，该防火墙可执行该策略。

经由该防火墙而加以执行的这项策略由一个或多个滤波器来定义。每个滤波器包括滤波器参数和关联的动作。这些滤波器参数被用来识别受制于该防火墙策略的网络信息包，并包括诸如硬件地址(例如，“媒体访问控制”(MAC)地址)、网络地址(例如，“网际协议”(IP)地址)、协议类型(例如，“传输控

制协议”(TCP))、端口号和类似物等信息。该动作定义应该如何处置具有与这些滤波器参数相匹配的参数的信息包。一个特殊的例子是：该滤波器包括“统一资源定位器”(URL)地址(例如，“http://www.foo.com”)，作为它的参数。该滤波器进一步使该阻滞动作(即，丢失该信息包)与那个URL地址相关联。只要该防火墙检查信息包并且通过那个检查而将该URL地址“http://www.foo.com”识别为被嵌入该信息包中，该服务器就丢失该信息包，从而防止它穿过网络。

利用通过包括分层网络结构的网络堆栈来发送和接收信息包，网络设备可以交换数据。存在不同的网络结构模型，但大多数至少包括应用层、传输层、网络层和链路层。网络信息包序贯地穿过每个层，并且，当每个层被穿过时，该信息包经历处理。关于出站信息包，该应用层根据应用协议(例如，其中的一些是：“超文本传输协议”(HTTP)、“文件传输协议”(FTP)和“简单邮件传输协议”(SMTP))来处理数据。其他的层(例如，该网络层和该传输层)通过将该数据嵌入TCP头部和IP头部中，来对其进行分组(packetize)。这些层通过(例如)分析头部、为数据解除分组(unpacketize)等，来为入站信息包执行互换处理。由这些层执行的该分层“堆栈”结构和处理功能会产生动态信息包结构，由此，当该信息包穿过该网络协议堆栈时，包括这些信息包参数的信息包内容会发生变化。

防火墙检查位于该分层网络堆栈内的检查点处的信息包。一方面，该检查点在该应用层处。例如，该防火墙被用作“分层服务供应者”(LSP)。该应用层处的信息包包括基础数据，该基础数据将被传送到另一个网络设备或者已从另一个网络设备那里被加以接收。通过检查该应用层处的该信息包，可允许该防火墙识别应用层参数(例如，URL 地址)并将这些应用层参数与这些滤波器参数进行比较。但是，其他的信息包参数(例如，IP 地址、端口号、MAC 地址和类似物)不可用，这是因为它们要么还没有被加入出站信息包，要么已离开入站信息包而被加以分析。

另一方面，在该网络堆栈(作为被置于该链路层与该网络层之间的中间驱动器)的较低层次处执行该防火墙检查点。该网络堆栈的这些较低层次处的信息包包括最大数量的参数(例如，接口号、MAC 地址、IP 地址、协议类型、端口和有效载荷数据)。虽然这些信息包包括这类参数，但是，这并不表示认为：可以容易地识别这些参数。在该防火墙接收该信息包之后，该防火墙需要分析

并解释该有关的信息包参数，用于和这些滤波器参数进行比较。这样，该网络堆栈中的这些层以及该防火墙可执行多余的信息包分析和解释功能。

发明内容

本发明针对一种在防火墙构架中执行防火墙的方法。该防火墙构架包括多个层，这些层中的每个层能够根据层协议来处理信息包，这些层中的每个层进一步能够请求将防火墙策略应用于这些信息包。该防火墙构架还包括防火墙引擎，该防火墙引擎包括多个被安装的滤波器。

请求层从这多个层中的前一层那里接收网络信息包。该请求层从该信息包中识别一组参数，并将分类调用发给具有该参数集的这个防火墙引擎。作为响应，该防火墙引擎将动作返回到该请求层。如果该动作是允许该信息包进一步穿过这多个层的指令，则该请求层根据该层协议来处理该信息包，并将该信息包发送到这多个层中的下一层。相反，如果该动作是丢失该信息包的指令，则该层不处理该信息包，并且不将该信息包发送到这下一层。

这个被安装的滤波器集中的每个滤波器包括一组滤波器条件和一个动作。当该请求层发出该分类调用时，该防火墙引擎确定：这多个被安装的滤波器中的任何滤波器是否与该参数集相匹配。然后，该防火墙引擎根据这些匹配滤波器中的信息来返回该动作。

本发明进一步提供了一种允许来自可信启动设备的信息包穿过防火墙(通常被配置成阻滞未经请求的入站信息包)并到达响应设备的方法。该启动设备首先启动使用密钥商议协议的鉴定。一旦成功地完成该鉴定，该响应计算机就创建一个滤波器，该滤波器允许来自该启动计算机的信息包到达该响应计算机处的目标处理。

通过以下参照附图而进行的对说明性实施例的详细描述，本发明的额外的特点和优点将会变得一目了然。

附图说明

所附权利要求书详细地陈述了本发明的这些特点，但通过以下结合附图的详细说明，可以最佳程度地理解本发明及其目的和优点。在这些附图中：

图 1 是框图，一般展示了其上驻留本发明的示范计算机系统；

图 2 是框图，一般展示了由此可使用本发明的示范网络环境；

图 3 是框图，一般展示了其中可使用本发明的各种方法的防火墙结构；

图 4 是框图，展示了用于本发明的示范滤波器；

图 5 是框图，展示了关于用于本发明的信息包上下文的示范数据结构；

图 6 是框图，展示了用于本发明的应用编程接口的示范集合；

图 7 是框图，展示了用于本发明的示范应用编程接口；

图 8 是框图，展示了根据本发明的、由网络层执行的各项功能；

图 9 是框图，展示了用于本发明的呼出(callout)的示范集合；

图 10 是流程图，展示了根据本发明的、被用来执行防火墙的示范方法；

图 11 是流程图，展示了被网络层用来执行防火墙的示范方法；

图 12 是流程图，展示了被防火墙引擎用来执行防火墙的示范方法；

图 13 是流程图，展示了一种方法，该方法被用来允许通过由可信网络设备启动的防火墙来进行未经请求的通信。

具体实施方式

描述了一种用于根据网络设备中的防火墙结构来执行防火墙的方法。该方法使网络信息包隶属协议堆栈中的多个层处的滤波器。在本发明的一个实施例中，在多个操作系统处理(被称作“核心模式处理”和“用户模式处理”)中执行该方法和防火墙结构。作为选择，在单一操作系统处理中，或者在在该操作系统以外执行的一个或多个程序模块或应用程序中执行该方法和结构。

该核心模式处理包括协议堆栈、核心防火墙引擎以及一个或多个呼出。该协议堆栈包括应用层、传输层、网络层和链路层。按需要增加或从该系统中删除额外的层。这些层每个都构成请求层，该请求层从前一个层或处理那里接收该网络信息包和对应的信息包上下文数据。然后，该请求层经由层 API，将分类请求发给该核心防火墙引擎。该分类请求包括由该请求层接收的该信息包、该信息包上下文和与该请求层关联的一组层参数。该核心防火墙引擎处理该请求，并返回动作。举例来讲，该动作指示该请求层如何处置该信息包(例如，允许或阻滞)。如果该动作是允许，则该请求层根据层协议来处理该信息包，将该信息包上下文修改成包括这些层参数，并将该信息包和信息包上下文传递到下一层。如果该动作是阻滞，则该请求层丢失该信息包，并且不将该信息包传递到这下一层。作为该阻滞动作的结果，该请求层可能会执行额外的功能(例如，拆毁 TCP 连接)。

该核心防火墙引擎包括该层 API、一组被安装的滤波器和呼出 API。这个被安装的滤波器集中的每个滤波器包括一组滤波器条件和一个关联的动作。该核心防火墙引擎通过识别一个或多个匹配滤波器，来处理从该请求层那里被发送的该分类请求。这些匹配滤波器具有跟这些层参数和信息包上下文相匹配的滤波器条件。一旦识别这些匹配滤波器，就按滤波器优先级的顺序来应用它们。如果正在被应用的该过滤的这个动作是允许或阻滞，则将这个动作返回到该请求层。如果该动作是呼出，则将该请求层所发出的该分类请求连同匹配滤波器标识一起传递到这些呼出模块中的一个呼出模块。该呼出模块执行其被编程的功能，并将动作返回到该核心防火墙引擎。如果没有为信息包识别匹配滤波器，则向该请求层通知：不曾发现匹配滤波器；然后，该请求层决定如何处置该信息包。

示范用户模式处理包括用户模式防火墙引擎以及一个或多个策略供应者。这些策略供应者从任何合适的来源(例如，易失或非易失存储器)获得策略。该策略是用于呈现新的滤波器的信息源，包括滤波器条件和关联动作的这个集合。该用户防火墙引擎经由滤波器引擎 API，将这个新的滤波器加入该核心防火墙引擎中的这个被安装的滤波器集。

该用户模式也包括该核心防火墙引擎的实例，从而允许创建用户模式层。然后，这些用户模式层使用该核心防火墙引擎的该用户模式实例来识别与一组参数相匹配的滤波器，这组参数允许在该用户模式以内应用过滤。

在本发明的实施例中，从该核心防火墙引擎到一组呼出模块的呼出接口实质上允许这些防火墙性能的无限扩展。举例来讲，HTTP 上下文呼出通过识别可接受的和不能接受的 URL 地址，来提供双亲特点。“网际安全性”(IPSec)呼出验证：信息包一直适当地经历 IPSec 处理。记录呼出记入符合所建立的标准的信息包，从而促进以后对信息包的检查。侵入检测呼出根据已知的算法来识别可疑的信息包。

本发明也提供了一种方法，用于允许与可信的网络设备进行未经请求的通信，同时阻滞来自其他网络设备的其他未经请求的通信。

参考这些附图(其中，相似的参考数字提及相似的元件)，本发明被展示为在合适的计算环境中加以执行。虽然未作要求，但是，将在正由个人计算机执行的计算机可执行指令(例如，程序模块)的一般上下文中描述本发明。通常，程序模块包括执行特殊任务或实施特殊的抽象数据类型的例行程序、程序、对

象、部件、数据结构等。也可以在分布式计算环境中实践本发明，在这些分布式计算环境中，由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中，程序模块可以位于本地记忆存储设备和远程记忆存储设备中。

图 1 展示了可以在其上执行本发明的合适的计算系统环境 100 的例子。计算系统环境 100 只是合适的计算环境的一个例子，它并不意在对本发明的使用或功能性的范围提出任何限制。也不应该将计算环境 100 解释为具有涉及示范操作环境 100 中所展示的任何一个部件或部件组合的任何从属性或要求。

本发明可用于众多其他的通用或专用计算系统环境或配置。可能适用于本发明的众所周知的计算系统、环境和/或配置的例子包括(但不局限于)个人计算机、服务器计算机、手持设备或便携式设备、多处理器系统、基于微处理器的系统、置顶盒、可编程的消费电子设备、网络 PC、小型计算机、大型计算机、包括以上任何系统或设备的分布式计算环境，以及类似物。

可以在正由计算机执行的计算机可执行指令(例如，程序模块)的一般上下文中描述本发明。通常，程序模块包括执行特殊任务或实施特殊的抽象数据类型的例行程序、程序、对象、部件、数据结构等。也可以在分布式计算环境中实践本发明，在这些分布式计算环境中，由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中，程序模块可以位于包括记忆存储设备的本地计算机存储介质和远程计算机存储介质中。

参照图 1，用于执行本发明的示范系统包括采取计算机 110 的形式的通用计算设备。计算机 110 的部件可以包括(但不局限于)处理单元 120、系统存储器 130 和系统总线 121，系统总线 121 将包括该系统存储器的各种系统部件耦合到处理单元 120。系统总线 121 可以是几种类型的总线结构(包括存储总线或存储控制器、外围总线和使用各种总线构造中的任何总线构造的局域总线)中的任何总线结构。举例来讲(不作限制)，这类结构包括“工业标准结构”(ISA)总线、“微通道结构”(MCA)总线、“增强的 ISA”(EISA)总线、“视频电子标准协会”(VESA)局域总线和也被称作“中层楼(Mezzanine)总线”的“外围部件互连”(PCI)总线。

计算机 110 通常包括各种计算机可读介质。计算机可读介质可以是可由计算机 110 存取的任何可用介质，它包括易失和非易失介质、可移动和不可移动的介质。举例来讲(不作限制)，计算机可读介质可以包括计算机存储介质和通信介质。计算机存储介质包括易失和非易失的可移动和不可移动的介质，这些

介质用关于信息(例如，计算机可读指令、数据结构、程序模块或其他数据)存储的任何方法或技术来加以执行。计算机存储介质包括(但不局限于)RAM、ROM、EEPROM、快闪存储器或其他存储技术、CD-ROM、数字通用光盘(DVD)或其他光盘存储器、盒式磁带、磁带、磁盘存储器或其他磁性存储设备、或可以被用来存储该所需信息并可以由计算机 110 来进行存取的其他任何介质。通信介质通常具体表现计算机可读指令、数据结构、程序模块或调制数据信号(例如，载波或其他传送机制)中的其他数据，它包括任何信息传递介质。术语“调制数据信号”意味着一种信号，该信号的一个或多个特征按这样的方式来加以设置或更改，以便为该信号中的信息编码。举例来讲(不作限制)，通信介质包括有线介质(例如，有线网络或直线连接)和无线介质(例如，声音、RF、红外线和其他无线介质)。以上任何内容的组合也应该被包括在计算机可读介质的范围以内。

系统存储器 130 包括采取易失和/或非易失存储器(例如，只读存储器(ROM)131 和随机存取存储器(RAM)132)的形式的计算机存储介质。基本输入/输出系统 133(BIOS)通常被存储在 ROM 131 中，该基本输入/输出系统包含有助于在计算机 110 内的各个元件之间传送信息(例如，在启动期间)的这些基本例行程序。RAM 132 通常包含可立即由处理单元 120 存取并且/或者目前正由处理单元 120 进行操作的数据和/或程序模块。举例来讲(不作限制)，图 1 展示了操作系统 134、应用程序 135、其他程序模块 136 和程序数据 137。

计算机 110 也可以包括其他可移动/不可移动的易失/非易失计算机存储介质。只举例来讲，图 1 展示了从不可移动的非易失磁性介质读取或对其写入的硬盘驱动器 141、从可移动的非易失磁盘 152 读取或对其写入的磁盘驱动器 151，以及从可移动的非易失光盘 156(例如，CD ROM 或其他光学介质)读取或对其写入的光盘驱动器 155。可以被用于该示范操作环境中的其他可移动/不可移动的易失/非易失计算机存储介质包括(但不局限于)卡型盒式磁带机、快闪存储卡、数字通用光盘、数字录像带、固态 RAM、固态 ROM 和类似的存储介质。硬盘驱动器 141 通常通过不可移动的存储接口(例如，接口 140)而被连接到系统总线 121，磁盘驱动器 151 和光盘驱动器 155 通常由可移动的存储接口(例如，接口 150)连接到系统总线 121。

以上所讨论的和图 1 中所展示的这些驱动器及其关联的计算机存储介质为计算机 110 提供计算机可读指令、数据结构、程序模块和其他数据的存储。在

图 1 中，例如，硬盘驱动器 141 被展示为存储操作系统 144、应用程序 145、其他程序模块 146 和程序数据 147。注意，这些部件可以等同于或不同于操作系统 134、应用程序 135、其他程序模块 136 和程序数据 137。这里为操作系统 144、应用程序 145、其他程序模块 146 和程序数据 147 提供不同的号码，以展示：它们至少是不同的副本。用户可以通过输入设备(例如，键盘 162)和定点设备 161(通常被称作“鼠标”、“跟踪球”或“触垫”)，来将命令和信息输入计算机 110。其他输入设备(未示出)可以包括话筒、操纵杆、游戏垫、圆盘式卫星电视天线、扫描仪或类似的输入设备。这些和其他的输入设备经常通过被耦合到该系统总线的用户输入接口 160 而被连接到处理单元 120，但也可以由其他接口和总线结构(例如，并行端口、游戏端口或通用串行总线(USB))来加以连接。监视器 191 或其他类型的显示设备也经由接口(例如，视频接口 190)而被连接到系统总线 121。除该监视器以外，计算机也可以包括其他外围输出设备(例如，扬声器 197 和打印机 196)，这些外围输出设备可以通过输出外围接口 195 来加以连接。

计算机 110 可以在使用与一台或多台远程计算机(例如，远程计算机 180)的逻辑连接的联网环境中进行操作。远程计算机 180 可以是另一台个人计算机、服务器、路由器、网络 PC、对等设备或其他共同的网络节点，它通常包括以上相对于个人计算机 110 而描述的许多或所有这些元件，尽管图 1 中只展示了记忆存储设备 181。图 1 中所描绘的这些逻辑连接包括局域网(LAN)171 和广域网(WAN)173，但也可以包括其他网络。这类联网环境在办公室、企业范围的计算机网络、内联网和因特网中很普遍。

当被用于 LAN 联网环境中时，个人计算机 110 通过网络接口或适配器 170 而被连接到 LAN 171。当被用于 WAN 联网环境中时，计算机 110 通常包括调制解调器 172 或用于在 WAN 173(例如，因特网)上建立通信的其他装置。调制解调器 172(可能是内置的，也可能是外置的)可以经由用户输入接口 160 或其他合适的机制而被连接到系统总线 121。在联网环境中，相对于个人计算机 110 或其各个部分而描绘的程序模块可以被存储在该远程记忆存储设备中。举例来讲(不作限制)，图 1 将远程应用程序 185 展示为驻留在存储设备 181 上。将会理解：所示的这些网络连接起示范的作用，可以使用在各台计算机之间建立通信链路的其他装置。

在下文中，除非另有指示，将参照动作和一台或多台计算机所执行的操作

的符号表示来描述本发明。照此，将会理解：这类动作和操作(有时被称作“是计算机执行的”)包括用结构化形式来表现数据的电信号的该计算机的该处理单元所执行的操作。这项操作变换该数据或将其保存在该计算机的该存储系统中的各个位置，这样，可按精通该技术领域的人所熟悉的方式来重新配置或改变该计算机的这个操作。那里保存有数据的这些数据结构是存储器的物理位置，该存储器具有由该数据的格式定义的特定属性。但是，前述上下文中描述了本发明，而这并不意在起限制的作用，如精通该技术领域的人将会理解的，也可以在硬件中执行下文中所描述的这各种动作和操作。

现在，将参照图 2 来描述其中使用本发明的用于执行防火墙的方法的一种网络环境。该网络本质上起示范的作用，因为可在被耦合到任何网络配置的任何网络设备中执行本发明的该方法。该网络环境包括专用网 200 和公用网 202。专用网 200 和公用网 202 属于任何合适的类型(例如，局域网(LANs)、广域网(WANs)、内联网、因特网或其任何组合)。

该网络环境包括多个网络设备 204、206、208、210 和 212。网络设备 204、206 被耦合到专用网 200。网络设备 210、212 被耦合到公用网 202。网络设备 208 被耦合到专用网 200 和公用网 202，并在这两个网络之间提供接口。这些网络设备被耦合到使用任何合适的技术(例如，以太网、1394 或 802.11(b))的该公用网和专用网。这些网络设备被进一步加以执行，作为任何合适的计算设备(例如，个人计算机、服务器、手持设备、打印机、交换器、路由器、桥接器、转发器或类似的设备)。

网络设备 208 包括防火墙 214 以及一个或多个滤波器 216。防火墙 214 是根据一种防火墙结构(其中，利用根据本发明的这种方法)来加以执行的一个程序模块或一组程序模块。防火墙 214 检查在被耦合到专用网 200 的网络设备 204、206、208 与被耦合到公用网 202 的网络设备 210、212 之间被交换的网络信息包。在本发明的实施例中，防火墙 214 也检查从专用网 200 内的网络设备那里被发送并去往那些网络设备的在本地被指定的网络信息包。

防火墙 214 在网络设备 208 中被加以执行，以保护并控制在专用网 200 与公用网 202 之间被交换的网络信息流通量，这被称作“边缘防火墙”。作为选择，防火墙 214 在如网络设备 210 中所展示的单一网络设备中被加以执行，并保护该单一网络设备，这被称作“主机防火墙”。该防火墙也能够按同步方式、作为主机和/或边缘防火墙的中央管理集来加以执行，这被称作“分布式防火

墙”。较佳地选择执行防火墙 214 的这个或这些网络设备的布置，以便防火墙 216 检查它应该保护的、为这些网络设备而指定的所有网络信息流通量。

滤波器 216 作为防火墙 214 的一部分来加以执行。作为选择，滤波器 216 作为防火墙 214 可以使用的单独的数据结构的一部分来加以执行。防火墙 214 和滤波器 216 执行一种防火墙策略，该防火墙策略被设计成保护网络设备 204、206、208 避免遭遇来源于被耦合到该公用网的网络设备 210、212 的恶意攻击。防火墙 214 也提供增加的功能性(例如，促进双亲控制、侵入检测、记录网络信息包和其他基于增加的滤波器的功能性)。

每个滤波器 216 包括一组滤波器条件以及一个或多个关联的动作。这些滤波器条件包括参数和信息，这些参数和信息可以从网络信息包(例如，接口号、硬件地址、网络地址、协议类型、端口号和有效载荷数据)中被加以分析或从网络信息包中获得。这一个或多个关联的动作定义执行该防火墙的该网络设备应该如何处置与这些滤波器条件相匹配的信息包。典型的动作包括允许(即，允许该信息包继续网络遍历)和阻滞(即，通过丢失信息包，来阻滞进一步的网络遍历)。

防火墙 214 检查当在网络设备 208 处被接收时穿过网络的网络信息包；并且，通过将这些信息包参数与这些滤波器条件进行比较，防火墙 214 识别一个或多个匹配滤波器。当这些滤波器条件与这些信息包参数相匹配时，产生匹配滤波器。和滤波器条件一样，这些信息包参数包括从信息包中被加以分析或获得的信息。当该防火墙识别匹配滤波器时，执行与这些滤波器条件关联的这一个或多个动作。

如这里所使用的该术语“信息包”指的是数据。该信息包可能是根据网络协议而加以格式化的网络信息包，也可能是由层、程序或模块处理的数据流。

图 3 表现了其中可执行本发明的该方法的一种防火墙结构的示范实施例。该方法提供了使信息包隶属于网络堆栈的所有层处的滤波器的能力。该方法提供了中央管理性能，这些中央管理性能允许增加和删除滤波器，并虑及待识别和解决的滤波器冲突。该防火墙结构是可扩展的，这体现在：滤波器层按需要来被增加和删除，并可以扩大，以包括允许和阻滞动作以外的专用功能性。虽然具体参照防火墙和防火墙滤波器来描述本发明，但是，也使用该方法来促进和管理其他的滤波器和策略。作为特殊的例子，本发明适合用来促进和管理被用于“服务质量”(QOS)、“网际协议安全性”(IPSec)套组以及其他的加密协

议、鉴定协议和密钥管理协议的滤波器。

该防火墙结构包括用户模式处理 250 和核心模式处理 252。用户模式处理 250 和核心模式处理 252 作为网络设备中的操作系统的一部分来加以执行。精通该技术领域的人将会理解：该操作系统的用户模式处理 250 和核心模式处理 252 包括额外的部件，为简单起见，没有示出这些额外的部件。作为选择，在作为一个或多个程序模块或应用程序的该操作系统以外或在单一操作系统处理以内整体或部分地执行该防火墙结构。

核心模式处理 252 包括网络堆栈 254、核心防火墙引擎 256 和任选的呼出 258。总的来说，核心模式处理 252 通过为网络信息包识别匹配滤波器、根据已知的协议来处理信息包并对如匹配滤波器所指定的该信息包执行其他动作，来实施被建立的防火墙策略。

网络堆栈 254 包括多个层，这些层包括数据流层 268、传输层 270、网络层 272 和链路层 274。该防火墙结构是可扩展的，并且，可按需要来动态地增加和除去额外的层。增加的层的例子包括文件存取层 276，该文件存取层根据“服务器主块”(SMB)协议来加以执行。这些层可以与其他程序模块(例如，“超文本传输协议”(HTTP)分析程序模块 278)协同运作。

网络堆栈 254 中的这些层处理入站网络信息包和出站网络信息包。出站网络信息包是正从执行该防火墙结构的该网络设备那里被传送到网络上的信息包。入站信息包是在执行该防火墙结构的该网络设备处被接收到的信息包。如图 3 中所示的这各个箭头所指出的，入站信息包从下到上地穿过网络堆栈 254，出站信息包从上到下地穿过网络堆栈 254。

网络信息包序贯地穿过这些网络层，并由这些网络层序贯地加以处理。根据已知的技术，网络堆栈 254 中的每个层能够从前一个层或模块那里接收信息包，能够根据规范或协议来处理该信息包，并能够将这个处理过的信息包发送到下一个层或模块。根据本发明，网络堆栈 254 中的每个层也保持信息包上下文，将该信息包上下文传递到这下一层，将分类请求发给核心防火墙引擎 256，并根据该防火墙策略来对该信息包采取行动。

该信息包上下文是从一个层到另一个层地跟随该信息包的数据结构。每个层通过将该层被设计成要加以处理的一组参数(例如，该层被设计成从信息包中加入、分析或导出的信息)加入该上下文数据结构，来保持该上下文。参照图 5 来描述被用于该信息包上下文的示范数据结构。

由网络堆栈 254 的各个层根据本发明来执行的这些操作中的一项操作是：通过发出该分类请求，来调用核心防火墙引擎 256。该分类请求是网络堆栈 254 中的层所执行的调用，请求：识别与该信息包相匹配的任何滤波器，并返回任何关联的策略(例如，防火墙策略)。发出该分类请求的这个层在这里被称作“请求阶段”或“请求层”。每个层也对由核心防火墙引擎 256 返回的该信息包采取该行动。用户模式层也可以构成请求层。

核心防火墙引擎 256 包括层 API 280、一组被安装的滤波器 282 和呼出 API 284。核心防火墙引擎 256 根据本发明的该方法来执行各种功能，包括：(1)保持定义该防火墙策略的被安装的滤波器集 282；(2)从网络堆栈 254 中的这些层接收分类请求；(3)根据该分类请求，来识别一个或多个匹配滤波器；以及(4)向该请求层指示将要被应用于该信息包的任何策略。

这个被安装的滤波器集中的每个滤波器包括一组滤波器条件以及一个或多个关联的动作。如参照图 2 的描述，这些滤波器条件识别经历这个关联的滤波器动作的这些网络信息包。被安装的滤波器集 282 中所规定的这些动作包括允许和阻滞。经由任选的呼出 258 来增加额外的功能性。参照图 4 来描述这些滤波器的示范形式。

层 API 280 在网络堆栈 254 中的各个层与核心防火墙引擎 256 之间提供接口。通过层 API 280，该请求层将该分类请求发给核心防火墙引擎 256。该分类请求包括如请求层所接收的该信息包、如该请求层所接收的该信息包上下文，以及层参数。这些层参数是由该请求层处理(例如，增加或分析)的信息包参数。作为特殊的例子，源网际协议(IP)地址和目的网际协议(IP)地址是当执行该 IP 协议时由网络层 272 发送的层参数。层参数也可以包括被加入该信息包或从该信息包中被加以分析的这些信息包参数以外的信息。作为特殊的例子，这些层参数包括局部地址类型。该局部地址类型由该 IP 层来确定，并作为该分类请求的一部分来加以发送。局部地址类型包括单点传送、广播、多点传送、任意点传送(anycast)和类似的类型。参照图 6 来描述层 API 280 的特殊实施。

随意地说，使用呼出 258 来执行该允许和阻滞滤波器动作以外的增加的功能性。当核心防火墙引擎 256 为该信息包(作为该关联的动作，它包括对这些呼出模块之一的呼出)识别匹配滤波器时，执行呼出。该核心防火墙引擎将如该请求层所发出的该分类请求(即完全信息包、层参数和信息包上下文)连同该

匹配滤波器的标识一起经由呼出 API 284 发送到该呼出模块。该防火墙结构包括呼出基本集 258。和层一样，按需要来增加额外的呼出，从而提供可扩展的结构。参照图 6 来描述呼出 API 284 的特殊实施。

用户模式处理 250 包括用户防火墙引擎 260，以及被识别为“PP1”、“PP2”和“PP3”的一个或多个策略供应者 262。策略供应者 262 是将防火墙策略(即被安装的滤波器 282)加入该防火墙结构的处理。可以使用任何处理来完成这项任务。一个例子是遗产(legacy)IPSec 策略服务(LIPS)。该遗产 IPSec 策略服务增加对应该使用 IPSec 协议(例如，“封装安全协议”(ESP)和“鉴定头部协议”(AH))的网络信息流通量进行定义的滤波器。作为特殊的例子，该遗产 IPSec 策略服务增加防火墙策略，该防火墙策略指出：必须根据该 ESP 协议，来为所有未经请求的入站信息包加密。该策略进一步规定：应该阻滞明文中的任何未经请求的入站信息包(即未加密的信息包)。策略供应者 262 从任何合适的来源(例如，易失或非易失存储器中的数据，或允许网络管理员或系统用户直接输入策略的“图形用户界面”(GUI))获得该策略。用户防火墙引擎 260 将该策略转换成新的滤波器，即，按照滤波器条件来定义该策略，并将这个新的滤波器加入被安装的滤波器集 282。

用户防火墙引擎 260 也执行滤波器判优和冲突解决功能。当策略供应者 262 为用户模式防火墙引擎 260 提供新策略时，该用户防火墙引擎确定：起因于这项新策略的这个新的滤波器是否与被安装的滤波器 282 中的任何滤波器有冲突。如果存在冲突，则用户防火墙引擎 260 解决该冲突。标题为《用于管理基于网络滤波器的策略的方法》的美国专利申请(代理人标签号：221037)中描述了一种适用于本发明的该构架中的、用于识别和解决冲突的方法的例子。

该结构进一步包括滤波器引擎 API 266，该滤波器引擎 API 266 构成用户模式防火墙引擎 260 与核心防火墙引擎 256 之间的接口。滤波器引擎 API 266 提供关于用户防火墙引擎 260 的机制，用于将新的滤波器加入被安装的滤波器集 282，并用于检查被安装的滤波器 282，以便可以检测和解决滤波器冲突。管理 API 290 向策略供应者 262 揭露了滤波器引擎 API 266 的功能性。

用户模式防火墙引擎 260 也包括滤波器模块 294。滤波器模块 294 是用户模式 250 中的核心防火墙引擎 256 的实例。用户模式防火墙引擎 260 中的滤波器模块 294 的这个实例允许用户防火墙引擎 260 为一个或多个用户模式层 282 复制核心防火墙引擎 256 的各项服务。用与创建核心模式层相同的方法来增加

用户模式层 282。由于滤波器模块 294 是核心防火墙引擎 256 的该用户模式实例，因此，将会理解：这里为该核心模式防火墙引擎而描述的任何功能性也可应用于滤波器模块 294。例如，增加额外的用户模式层或从该系统结构中删除额外的用户模式层，并可以创建呼出，从而为这些用户模式层提供增加的功能性。

密码模块 API 288 在用户策略引擎 260 与密码模块 296 之间提供接口。该密码模块提供了一种机制，用于确定为给定的信息包使用哪些安全设置。使用密码模块 API 288 来用信号通知该密码模块：需要建立 SA。

参考图 4，现在将描述被安装的滤波器集 282。每个滤波器 310 具有多个字段，这些字段包括滤波器 Id 312、加权 314、一个或多个动作 316、策略上下文 317 和一组滤波器条件 318。滤波器 Id 312 为该滤波器提供唯一标识。例如，滤波器 Id 312 被用作供核心防火墙引擎 256 将匹配滤波器信息返回到用户防火墙引擎 260 和呼出 258 的一种工具。在本发明的实施例中，滤波器 310 被分配给网络堆栈 254 中的各个层中的一个层。滤波器 Id 312 被核心防火墙引擎 256 用来跟踪将哪个滤波器分配给哪个层。

加权字段 314 包括识别滤波器 310 的优先级的值。加权字段 314 中的这个值越大，该滤波器的优先级就越高。该滤波器优先级确定核心防火墙引擎 256 将匹配滤波器应用于该信息包的那个顺序。

在本发明的实施例中，首先应用具有最高优先级(即最大的加权值)的那个滤波器，然后应用优先级位于其次的滤波器，等等，直到遇到具有终止动作的匹配滤波器为止。以下更加详细地描述了终止动作。一旦应用具有该终止动作的这个匹配滤波器，核心防火墙引擎 256 就停止应用匹配滤波器。这样，在应用该终止动作之后，不对该信息包采取由较低优先级匹配滤波器规定的动作 316。作为选择，防火墙引擎 256 识别单一匹配滤波器，并返回来自该单一匹配滤波器的一组动作。不管加权值 314 如何，防火墙引擎 256 也可以应用所有的匹配滤波器。

滤波器条件集 318 确定信息包是否与滤波器 310 相匹配。每个滤波器条件 318 包括类型 320、数据 322 和层 Id:字段 Id 324。类型 320 定义对应的数据 322 中所包括的变量的长度和数目。该结构规定预定义的已知变量类型(例如，“字节”、“短”、“长”、“8 个字节”、“字符串”、“网际协议版本 4(IPv4)地址”、“网际协议版本 6(IPv6)地址”、“IPv4 地址加上掩码”、“IPv6 地

址加上掩码”和“地址范围”)。数据字段 322 包括与该类型相匹配的数据。例如，如果该类型是“IPv4 地址”，则关于数据字段 322 的可接受的值是如有点的十进制记数法中所表达的 00.00.00.00~255.255.255.255 的范围内的 32 位数字。在一些实例中，类型 320 规定数据字段 322 中的多个值。“地址范围”、“IPv4 地址加上掩码”和“IPv6 地址加上掩码”这些类型允许采用两个 IP 地址值，从而定义 IP 地址的始、末范围。关于最大的灵活性，该结构也允许采用用户定义的类型。作为选择，将额外的类型手动加入该系统结构。

分别使用层 Id:字段 Id 332 来分别识别起源层和来自该起源层的参数。该起源层和来自该起源层的这个参数定义信息包参数，即与数据 322 进行比较的层参数和信息包上下文。该起源层识别该网络堆栈中的层。来自该起源层的这个参数识别与该起源层关联的特殊参数。滤波器条件 326 展示了特殊的例子。该类型是 IPv4，从而指出：数据 322 是 32 位 IP 地址。该层 Id 是“IP”，表示：该 32 位数字是 IP(即网络)层参数。该字段 Id 是“Src IP Addr”，它在该例中代表 IP 层参数(尤其是源 IP 地址)。该数据字段中所提供的该源 IP 地址是“123.3.2.1”，指出：具有那个源 IP 地址的任何信息包都符合该滤波器条件，从而与该滤波器相匹配。

可以规定多个滤波器条件 318。当规定多个滤波器条件 318 时，并且当满足所有的滤波器条件 318 时，信息包与滤波器 310 相匹配。

滤波器 310 中被指定的动作 326 是允许、阻滞或呼出。如果滤波器 310 中的动作 324 是允许或阻滞，并且该信息包与滤波器 310 相匹配，那么，该允许或阻滞动作被核心防火墙引擎 256 返回到该请求层。如果动作 316 是呼出，则核心防火墙引擎 256 将其自己的分类请求发给指定的呼出模块 258，该分类请求包括该完全信息包、层参数、上下文和该匹配滤波器的标识。呼出模块 258 对该信息包执行其被编程的功能(例如，侵入检测)。该呼出可以将动作(允许或阻滞)返回到该核心防火墙引擎，这又将该动作转达给该请求层。该呼出也可以指示该核心防火墙引擎继续应用信息包，而不会提供允许或阻滞动作。该呼出也能够保持信息包上下文，该信息包上下文同样经由核心防火墙引擎 256 而被返回到该请求层。

动作被指定为终止或非终止。默认的情况是，“允许”和“阻滞”被指定为终止动作。终止动作是这样一种动作——一旦在匹配信息包中被识别，它就可以被用来停止如前所述的应用匹配滤波器的过程。

使用策略上下文 317 来存储除防火墙策略以外的策略(例如，安全性策略或 QOS 策略)。该策略上下文是任何合适的数据结构。例如，该策略上下文是由加入过该策略上下文的处理来解释的 64 位数字。该策略上下文和/或动作也可能是空值。

图 5 展示了被用于信息包上下文的数据结构 330 的例子，该信息包上下文由网络堆栈 254 和呼出模块 258 中的各个层来保持，并被传递到这些层。当这个入站或出站网络信息包穿过这些层并包括被标注为 336-340 的一个或多个项目时，信息包上下文 330 跟随该网络信息包。每个项目包括层 Id:字段 Id 332 和对应的值 334。

层 Id:字段 Id 332 的含义等同于作为滤波器 310 中的滤波器条件 318 的一部分而加以提供的层 Id:字段 Id 324 的含义(图 4)。也就是说，层 Id:字段 Id 322 为值字段 334 中的该数据识别该起源层和来自该起源层的这个层参数。值字段 334 包括特殊的层参数。

作为特殊的例子，项目 336 包括该层 Id:字段 Id 332“NDIS: Src.MAC Addr.”。“NDIS”表示链路层 274 的“网络驱动器接口规范”实施(图 1)。“Src MAC addr.”表示源 MAC 地址。这样，层:字段 Id 332 指出：值字段 334 中的该数据是曾由该 NDIS(链路)层处理的源 MAC 地址。值字段 334 包括这个实际的源 MAC 地址，在该例中，该源 MAC 地址是如十六进制记数法中所表达的“00.08.74.4F.22.E5”。

作为第二个例子，项目 338 具有“NDIS:IF No”的层 Id:字段 Id 332。这再次将该层识别为 NDIS，但在此情况下，将该参数识别为把接口号表示为该特殊 NDIS 参数的“IF No”。值字段 334 包括这个实际的接口号，该接口号在此情况下是 2。

作为第三个例子，项目 340 具有“IP:Dst IP Addr”的层 Id:字段 Id 332。该“IP”使用该 IP 协议来表示该网络层，并且，该“Dst IP Addr”将目的 IP 地址表示为该 IP 层参数。值字段 334 包括“123.3.2.1”的这个实际的目的 IP 地址。

已描述了该基础防火墙结构，现在来注意使用这里所描述的该基础防火墙结构来加以执行的该系统和示范方法的这些功能接口。这些功能接口作为多个应用编程接口(APIs)来加以执行。这些 APIs 包括如图 6 和图 7 中作为例证所示的层 API 280、呼出 API 284、滤波器引擎 API 266 和密码模块 API 288。

层API 280促进网络堆栈254内的各个层中的每个层与核心防火墙引擎256之间的数据交换。如所示，层API 280包括“分类”法350、“增加层”法352和“删除层”法354。

该请求层使用“分类”法350，将层参数、如该请求层所接收的该信息包以及该信息包上下文发送到核心防火墙引擎256。核心防火墙引擎256将(1)来自该请求层的这些层参数和(2)信息包上下文项目跟被分配给该请求层的每个滤波器310中的滤波器条件318进行比较，以识别匹配滤波器。以下是该“分类”法的示范实施。将会理解：以下这些方法被描述成接收或返回数据值。根据已知的编程技术，这些方法可以使用指向数据值的指针，而不是实际的数据值。

```
NTSTATUS
WFPClassify
(
        IN ULONG                                LayerId,
        IN WFP_INCOMING_VALUES*                 pInFixedValues,
        IN WFP_INCOMING_CONTEXT_VALUE*          pInContext,
        PVOID                                   pPacket,
        OUT WFP_ACTION_TYPE*                    pActionType,
        OUT UINT64*                             pOutContext
);
```

其中，以下内容表现了这些所列举的参数的特征。

LayerId 识别发出该分类请求的该网络层，即该请求层。参考图3，该层Id将该层识别为数据流层268、传输层270、网络层272或链路层274。如果被加入该系统，则其他层(包括用户模式层)有效。例如，如果加入SMB层276，则该SMB层具有其自己的唯一标识。本发明的该防火墙结构进一步允许网络堆栈254中的层处的多项协议实施。例如，该堆栈具有两个传输层270——第一传输层使用该TCP协议，第二传输层使用该UDP协议。

pInFixedValues 包括由该请求层处理的这些层参数的子集。将该pInFixedValues连同这些信息包上下文项目与这些滤波器条件进行比较，以确定该信息包是否与该滤波器相匹配。以下的表格 A 中识别关于每个层的该pInFixedValues中所包括的默认层参数。将会理解：这些默认层是例子，而不起限制的作用，因为该层可以包括它在该 pInFixedValues 中可以使用的任何参数。

表格 A

层	默认层参数
链路层	源 MAC 地址和目的 MAC 地址； 接口号
网络层	源 IP 地址和目的 IP 地址； 协议类型；局部地址类型
传输层	源端口号和目的端口号；
应用程序	被解密的应用层协议有效载荷

pInContext 包括如该请求层所接收的上下文数据结构 330(图 5)。核心防火墙引擎 256 使用该信息包上下文并结合这些层参数来识别匹配信息包。

pPacket 包括如该请求层所接收的这整个信息包。该 pPacket 没有被核心防火墙引擎 256 用来识别匹配滤波器。如前所述，核心防火墙引擎 256 使用该 pInFixedValues 和 pInContext 来识别匹配滤波器。该 pPacket 被包括在该“分类”法中，以便核心防火墙引擎 256 可以将它发送到被识别为匹配滤波器中的动作 316 的一个或多个呼出模块 258。

pActionType 包括被返回到该请求层的动作 316。所返回的动作 316 是如在该匹配滤波器中所识别的允许、阻滞或无、或由该匹配滤波器执行的呼出模块。

pOutContext 包括该策略上下文数据。如前所述，使用该策略上下文来推动跟 IPSec、QOS 和任何其他基于非防火墙滤波器的策略有关联的网络策略。

“增加层”352 和“删除层”354 方法被分别用来增加层和从该防火墙结构中除去层。以下是“增加层”352 方法的示范形式。

```
NTSTATUS
AddExtensionLayer(OUT PULONG pLayerId);
```

其中，以下内容表现了这些所列举的参数的特征。

pLayerId 是被返回到正在添加的那个层(即执行该“增加层”法的那个层)的唯一层标识值。

以下是“删除层”406 方法的示范形式。

```
NTSTATUS
RemoveExtensionLayer(ULONG LayerId);
```

其中，以下内容表现了这些所列举的参数的特征。

LayerId 识别正在被除去的那个层，即执行该“删除层”法的那个层。

呼出 API 284 促进核心防火墙引擎 256 与呼出 258 之间的数据交换。和层 API 280 一样，“呼出”API 284 具有“分类”法。“呼出”API 284 的“分类”法 356 类似于“层”API 280 的“分类”法 350，除“它也包括匹配滤波器数据”这一点以外。以下是被用来执行呼出的“分类”法 356 的示范形式。

```
typedef NTSTATUS (*WFP_CALLOUT_CLASSIFY_FN)
(
        IN const WFP_INCOMING_VALUES*          fixedValues,
        IN WFP_INCOMING_CONTEXT_VALUE*         wfpContext,
        IN VOID*                               packet,
        IN WFP_FILTER*                         matchedFilter,
        OUT WFP_ACTION_TYPE*                   action,
        OUT UINT64*                            outContext
);
```

其中，以下内容表现了这些所列举的参数的特征。

fixedValues 包括从该请求层那里被发送的这些层参数。该 fixedValues 是 pInFixedValues 数据中的该请求层所提供的相同的数据，该 pInFixedValues 数据作为层 API 280 中的“分类”法 350 的一部分来加以发送。

wfpContext 包括上下文数据结构 330(图 5)。该数据等同于如该 pInContext 中的该请求层所发送的数据，该 pInContext 作为层 API 280 中的“分类”法 350 的一部分来加以发送。

packet 包括如该请求层所接收的这整个信息包。该数据等同于如该 pPackett 中的该请求层所发送的数据，该 pPacket 作为层 API 280 中的“分类”法 350 的一部分来加以发送。

matchedFilter 识别请求该呼出的那个滤波器。通常，该匹配滤波器由启动呼出 API 284 的“分类”法 356 的匹配滤波器 310 的滤波器 Id 312 来加以识别。

pActionType 包括从呼出 258 被返回到核心防火墙引擎 256 的那个动作。如果该 pActionType 是允许或阻滞，则将它返回到该请求层，作为由“层”API 280 返回的该 pActionType。该呼出也可以返回继续动作，该继续动作指示核

心防火墙引擎 256 继续将匹配滤波器应用于该信息包。

pOutContext 包括该策略上下文数据(例如，安全性或 QOS 策略数据)。

呼出 API 408 也包括“通知”法 358。当滤波器 310 被加入被安装的滤波器集 282(将呼出模块 258 识别为其动作 316 之一)时，使用“通知”法 358 来通知呼出。该通知为该呼出提供了一种机会——采取任何所要求的动作(例如，当核心防火墙引擎 256 执行该动作时，分配或解除分配将由呼出 258 使用的缓冲器)。以下是“通知”法 358 的示范形式。

```
typedef NTSTATUS (*WFP_CALLOUT_NOTIFY_FN)
(
        IN WFP_NOTIFY_ENUM    notify,
        IN WFP_FILTER*        filter
);
```

其中，以下内容表现了这些所列举的参数的特征。

notify 包括一个数值，该数值指出是正在增加还是正在删除该滤波器。例如，1 的值指出正在增加该滤波器，2 的值指出正在删除该滤波器。

filter 识别正在被唯一值增加或删除的该滤波器。通过提供作为滤波器 310 的一部分而被包括在内的滤波器 Id 312，可以实现这一点。

该呼出 API 也包括“呼出登记”法 360 和“呼出解除登记”362 方法，以分别增加和除去呼出模块。“呼出登记”方法 360 的示范形式如下所述：

```
NTSTATUS WfpRegisterCallout
(
        IN const GUID*                  calloutId,
        IN const WFP_CALLOUT*           callout,
        IN const SECURITY_DESCRIPTOR*   sd
);
```

其中，以下内容表现了这些所列举的参数的特征。

callout Id 提供关于该登记呼出模块的唯一标识。

callout 提供任何呼出特定信息(例如，驱动器服务名称、设备名称，以及指向这些呼出分类和通知功能的指针)。

sd 提供关于该呼出的安全描述符。该安全描述符识别哪些处理可以读取和删除该呼出。

“呼出解除登记”法 362 的示范形式如下所述：

```
NTSTATUS WfpDeregisterCallout
(
        IN const GUID*      calloutId
);
```

其中，以下内容表现了这些所列举的参数的特征。

callout Id 是将要被除去的该呼出的唯一 Id。

滤波器引擎 API 266 促进用户模式防火墙引擎 260 与核心模式防火墙引擎 256 之间的数据交换。如所示，滤波器引擎 API 266 包括“增加滤波器”法 364、“删除滤波器”法 366 和“列举层”法 368。如前所述，滤波器引擎 API 266 的这些方法可以被包括在管理 API 290 中，以便向策略供应者 262 揭示那里的功能性。

分别使用“增加滤波器”364 和“删除滤波器”366 方法，来将新的滤波器加入被安装的滤波器集 282，并从被安装的滤波器集 282 中删除现存的滤波器。以下是“增加滤波器”法 364 的示范形式。

```
NTSTATUS
AddFilterToLayer
(
        ULONG               LayerId,
        WFP_FILTER*         pFilter
);
```

其中，以下内容表现了这些所列举的参数的特征。

LayerId 识别被分配给该滤波器的那个层。

pFilter 是正在被加入被安装的滤波器集 282 的滤波器 310。

以下是“删除滤波器”法 366 的示范形式。

```
NTSTATUS
DeleteFilterFromLayer
(
        ULONG       LayerId,
        ULONG       FilterId
);
```

其中，以下内容表现了这些所列举的参数的特征。

LayerId 识别为其分配该滤波器的那个层。

pFilter 是正从这个被安装的滤波器集中被删除的该滤波器。

“列举层”法 368 提供关于用户防火墙引擎 260 的一种机制，以识别与一套标准相匹配的所有滤波器。这样，可允许该滤波器引擎 API 识别发生冲突的滤波器，用于滤波器判优和冲突解决。以下是“列举层”法 368 的示范形式。

```
IndexStartEnum
(
        PWFP_ENUM_TEMPLATE      pEnumTemplate,
        OUT PULONG              pMatchCount,
        OUT PWFP_ENUM_HANDLE    pEnumHandle
)
```

其中，以下内容表现了这些所列举的参数的特征。

pEnumTemplate 包括定义将要被返回的这些滤波器的一种数据结构。例如，它包括参数，这些滤波器条件必须为将要被返回的该滤波器而与这些参数相匹配。

pMatchCount 包括基于这个规定的 pEnumTemplate 的滤波器匹配数目。

pEnumHanlde 包括对这些匹配的滤波器项目的参考。

密码模块 API 288 在用户模式密码模块层 282 与用户防火墙引擎 260 之间提供接口。密码模块 API 288 包括“IPSec SA 获取”法 370、“期满通知”法 372、“IPSec SA 获取完成”法 374、“密码模块登记”法 376、“密码模块解除登记”法 378、“IPSec 入站获得 SPI”法 380、“增加入站 SA”法 382、“增加出站 SA”法 384、“入站 SA 期满”386 方法和“密码模块启动”法 388。

使用该密码模块 API 来促进安全协议(例如，由 IPSec 定义，由启动计算机和响应计算机使用)的运用。IPSec 包括诸如鉴定头部(AH)和封装安全协议(ESP)等协议。该 ESP 协议(主要在“IETF 请求注解”(RFC)2406 中加以证明)是鉴定和加密协议，该协议使用密码机制来提供完整性、来源鉴定和数据机密性。该 AH 协议(主要由 IETF RFC 2402 来证明)是鉴定协议，该协议使用该信息包头部中的散列签名来证实该信息包数据的完整性和该发送者的可靠性。

该 IKE 协议(主要在 IETF RFC 2409 中加以证明)提供一种关于启动计算机和响应计算机的方法，以商议用于该 AH 协议和 ESP 协议的安全设置。所商议的这些安全设置构成被称作“安全关联”(SA)的数据结构。该 SA 定义被 ESP 或 AH 用来保护 IP 信息包的这些内容的参数(例如，鉴定算法、加密算法、密钥和密钥的使用期限)。由于 ESP 和 AH 需要被建立的 SA，因此，在该启动和响应计算机使用该 ESP 或 AH 协议之前，执行 IKE 商议。由被称作“安全参数索引”(SPI)的值来识别给定的 SA。

该启动计算机和响应计算机中的每种计算机包括 IPSec 驱动器，该 IPSec 驱动器根据 IPSec 策略来确定在该启动计算机与该响应计算机之间被发送的数据是要求加密还是要求鉴定。该 IPSec 策略是一组滤波器，它定义该网络设备

如何使用 IPSec，并包括滤波器清单、鉴定方法和其他信息。在本发明的实施例中，由这个被安装的滤波器集中所包括的滤波器来定义该 IPSec 策略。

用户防火墙引擎 260(经由客户代理人)调用“IPSec SA 获取”法 370，来将驱动器获取或外部启动请求传递到该密码模块。该密码模块层返回这个调用，并异步地执行该商议。一旦这个密码模块已完成该商议，该密码模块层就调用“IPSec SA 获取完成”法 374，以通知该用户防火墙引擎：该商议完成。以下是该“IPSec SA 获取”法的示范形式。

```
typedef WIN32_ERR
(*PROCESS_IPSEC_SA_ACQUIRE0)
(
        IN FWP_IPSEC_ACQUIRE_CONTEXT0   ipsecContext,
        IN const FWP_IPSEC_SA_ACQUIRE0* acquire,
        IN FWP_IPSEC_SPI                inboundSAspi
);
```

其中，以下内容表现了这些所列举的参数的特征。

ipsecContext 是将该获取与正在被加入的该 SA 连接起来的句柄。

acquire 包括用于根据已知协议(例如，IKE)来商议该 SA 的必要信息。

inboundSAspi 包括被用于入站 SA 的 SPI。

调用“期满通知”法 372，以便将期满通知传递到增加过该入站 SA 的那个密码模块。以下是该“期满通知”法的示范形式。

```
typedef VOID
(*PROCESS_IPSEC_SA_EXPIRE0)
        (
        IN const FWP_IPSEC_SA_EXPIRE_NOTIFY0* expireNotify
        );
```

其中，以下内容表现了这些所列举的参数的特征。

expireNotify 包含识别该期满 SA 的信息。例如，在出站 SA 的情况下，提供该 SPI。

密码模块调用“IPSec SA 获取完成”374 方法，以便在它已结束商议并增加所有这些 SA 之后，或在它已碰到错误之后，关闭该用户防火墙引擎的上下文。在执行该方法之后，该密码模块层没有再使用关于其他任何 API 方法的该 ipsecContext。以下是该“IPSec SA 获取完成”法的示范形式。

```
WIN32_ERR
FwpIPSecSAAcquireComplete0
(
        IN FWPM_ENGINE_HANDLE                          engineHandle,
        IN FWP_IPSEC_ACQUIRE_CONTEXT0                  ipsecContext,
        IN const FWP_IPSEC_NEGOTIATION_STATUS0*        status
        );
```

其中，以下内容表现了这些所列举的参数的特征。

engineHandle 为用户防火墙引擎 260 提供句柄。

ipsecContext 是由利用该“IPSec 获取”法的用户防火墙引擎传递的该上下文。

status 提供该 SA 商议的状态和其他细节。如果经由 FwpKeyingModuleInitiate0 在外部启动过该获取，则用户防火墙引擎 260 返回该状态。

由密码模块层来调用“密码模块登记”法 376，以便向用户防火墙引擎 260 登记，并传递其功能指针。以下是该“密码模块登记”法的示范形式。

```
WIN32_ERR
FwpKeyingModuleRegister0
(
        IN FWPM_ENGINE_HANDLE                  engineHandle,
        IN const GUID*                         keyingModuleID,
        IN const FWP_KEYING_MODULE_INFO0*      keymodInfo
);
```

其中，以下内容表现了这些所列举的参数的特征。

engineHandle 为用户防火墙引擎 260 提供该句柄。

keyingModuleID 是关于该密码模块的唯一 ID。

keymodInfo 包括关于该密码模块层的登记信息(例如，用于处理“IPSec SA 获取”和“IPSec SA 期满”功能的指针)。

由该密码模块来调用“密码模块解除登记”法 378，以便从用户防火墙引擎 260 中解除登记该密码模块。以下是该“密码模块解除登记”法的示范形式。

```
WIN32_ERR
FwpKeyingModuleDeregister0
(
        IN FWPM_ENGINE_HANDLE      engineHandle,
        IN const GUID*             keyingModuleID
);
```

其中，以下内容表现了这些所列举的参数的特征。

engineHandle 是给用户防火墙引擎 260 的该句柄。

keyingModuleID 是该密码模块层的唯一 ID。

由该密码模块层来调用“IPSec 入站获得 SPI”法 380，以获得关于新的入站 SA 的该 SPI。当该密码模块层在响应的网络设备中执行时，通常使用“IPSec 入站获得 SPI”法 380。以下是该“IPSec 入站获得 SPI”法的示范形式。

```
WIN32_ERR
FwpIPSecSAInboundGetSpi0
(
        IN FWPM_ENGINE_HANDLE              engineHandle,
        IN const FWP_IPSEC_TRAFFIC0*       ipsecTrafficDescription,
        IN const FWP_IPSEC_UDP_ENCAP0*     udpEncapInfo,
        OUT FWP_IPSEC_SPI*                 inboundSpi
);
```

其中，以下内容表现了这些所列举的参数的特征。

engineHandle 是给用户防火墙引擎 260 的该句柄。

ipsecTrafficDescription 是用于创建入站幼小状态的 SA 的 5 元组说明。该 5 元组包括源 IP 地址与目的 IP 地址、源端口与目的端口，以及传输层协议类型。

udpEncapInfo 是用于创建该幼小状态的 SA 的 UDP 封装数据。UDP 封装是将根据安全协议而加以格式化的信息包嵌入未加密的 UDP 信息包的一种已知的方法。

inboudSpi 是关于该入站 SA 的该 SPI。

由该密码模块层来调用“增加入站 SA”法 382，以增加入站 SA(即，更新幼小状态的 SA)。用户防火墙引擎 260 使用该 SA 中的该 SPI 来将此调用映射到其内部状态，并将该 SA ioctl down 到该 IPSec 驱动器。以下是该“增加入站 SA”法的示范形式。

```
WIN32_ERR
FwpIPSecSAInboundAdd0
(
        IN FWPM_ENGINE_HANDLE              engineHandle,
        IN const FWP_IPSEC_SA_STRUCT0*     inboundSA
);
```

其中，以下内容表现了这些所列举的参数的特征。

engineHandle 是给该用户防火墙引擎的该句柄。

inboundSA 包括该入站 SA。

由密码模块来调用“增加出站 SA”384 方法，以增加出站 SA。该用户防火墙引擎使用入站 SPI 参数来将此调用映射到其内部状态，并将该 SA ioctl down

到该 IPSec 驱动器。以下是“增加出站 SA”法的示范形式。

```
WIN32_ERR
FwpIPSecSAOutboundAdd0
(
        IN FWPM_ENGINE_HANDLE       engineHandle,
        IN FWP_IPSEC_SPI            inboundSpi,
        IN const FWP_IPSEC_SA_STRUCT0* outboundSA
);
```

其中，以下内容表现了这些所列举的参数的特征。

engineHandle 是给用户防火墙引擎 260 的该句柄。

onboundSpi 是关于该入站 SA 的该 SPI，该入站 SA 与出站 SA 配对。

outboundSA 包括该出站 SA。

由该密码模块来调用“入站 SA 期满”法 386，以截止先前增加的该入站 SA。以下是“入站 SA 期满”法 386 的示范形式。

```
WIN32_ERR
FwpIPSecSAInboundExpire0
(
        IN FWPM_ENGINE_HANDLE           engineHandle,
        IN const FWP_IPSEC_SA_EXPIRE0*  expire
);
```

其中，以下内容表现了这些所列举的参数的特征。

engineHandle 是给用户防火墙引擎 260 的该句柄。

expire 包括关于将要期满的该 SA 的数据。

由已知的外部应用程序(比如 RAS、Winsock API 和类似物)来调用“密码模块启动”法 388，以便在该应用程序开始发送其网络信息流通量之前，启动该密码模块层并设置 SA。用户防火墙引擎 260 异步地使该 RPC 调用待决，从 IPSec 驱动器那里获得该 SPI，并将该获取传递到这个合适的密码模块。一旦该密码模块层调用 FwpIPSecSAAcquireComplete0，该用户防火墙引擎就完成具有该商议状态的该异步 RPC。以下是该“密码模块启动”法的示范形式。

```
WIN32_ERR
FwpKeyingModuleInitiate0
(
        IN FWPM_ENGINE_HANDLE                   engineHandle,
        IN const FWP_IPSEC_SA_ACQUIRE0*         acquire,
        IN HANDLE                               waitEvent,
        OUT FWP_IPSEC_NEGOTIATION_STATUS0*      negotiationStatus
);
```

其中，以下内容表现了这些所列举的参数的特征。

engineHandle 是给用户防火墙引擎 260 的该句柄。

acquire 包括商议 SA 所必要的数据。

waitEvent 是给当存在商议状态时被触发的事件的句柄。如果客户(即，调用外部应用程序)没有兴趣等待该商议完成，则它可以将这个参数设置为 NULL。在内部，该客户代理人随意地将这个事件传递到 RPC，并且，一旦该异步 RPC 调用完成，就请求它设置这个事件。

negotiationStatus 包括该商议的结果。如果 waitEvent 是 NULL，则该 negotiationStatus 是 NULL。否则，negotiationStatus 保持有效，直到该 waitEvent 被触发为止。

图 8 展示了根据本发明的、由网络堆栈 254 中的这些层使用的各种方法。图 8 中所展示的该方法也可以由一个或多个用户模式层 282 跟滤波器模块 294 和用户防火墙引擎 260 协力来加以使用。

每个层能够执行多项功能，包括：处理网络信息包；将分类请求发给核心防火墙引擎 256；以及，管理信息包上下文。在本发明的实施例中，在网络堆栈 254 中所安装的薄垫片 400、402、404、406 内的各个层中的每个层处执行这些功能。作为选择，该功能性被直接置入这些单独的层，而不需要这些薄垫片。

网络堆栈 254 包括数据流层 268、传输层 270、网络层 272 和链路层 274。出于展示本发明的目的，链路层 274 作为 NDIS 驱动器来加以执行，网络层 272 作为 IP 层来加以执行，传输层 270 作为 TCP 层来加以执行，数据流层 268 作为 HTTP 层来加以执行。将会理解：可以根据任何协议来执行层。例如，该传输层也配合“用户数据报协议”(UDP)。该应用层支持“文件传输协议”(FTP)、“远程过程调用”(RPC)、“简单邮件传输协议”(SMTP)、“服务器主块”(SMB)等。如前所述，可以将额外的层加入该结构，并可以删除层。例如，使用参照图 6 来加以描述的该“增加层”法和“删除层”法，来增加和删除层。

被标注为 408(a)-(d)的网络信息包在它穿过网络堆栈 254 中的各个层并由这些层来处理时，展示了该网络信息包。如果信息包 408(a)-(d)是入站信息包，则它从下到上穿过该网络堆栈。如果信息包 408(a)-(d)是出站信息包，则它从上到下穿过该网络堆栈。这类处理众所周知，但出于展示本发明的目的，对其进行简要的描述。

假设在网络设备(例如，web 浏览器)中执行的应用程序启动关于位于另一

个网络设备上的网页的内容的请求，则该应用程序将该请求发给数据流层 268。在该例中，数据流 264 根据该 HTTP 协议来对该请求进行格式化，并将该请求发送到信息包 408(a)中的该传输层。传输层 270 接收信息包 408(a)。传输层 270(执行该 TCP 协议)将该数据放置在一个或多个信息包中，并且，为每个信息包提供 TCP 头部。该 TCP 头部包括诸如该源端口与目的端口、协议类型(即 TCP、序号、标记和检验和)等信息。然后，该传输层将被标注为 408(b)的该信息包发送到该网络层。

该网络层执行该 IP 协议，并将该数据装入 IP 头部，该 IP 头部包括该源 IP 地址与目的 IP 地址、标记、检验和以及其他已知信息。该 IP 头部也指出是否为该信息包分段。当该 IP 信息包的尺寸超过关于被用来传送信息包的该网络技术的“最大传输单位”(MTU)尺寸时，为该信息包分段。例如，以太网技术规定：该 MTU 是 1500 个字节。如果该 IP 信息包长度超过该 MTU，则将它分成两个或多个 IP 信息包—— 每个 IP 信息包具有其自己的 IP 头部，与该 MTU 相比，它们都具有相等的或较短的长度。

在本发明的实施例中，该网络层被分成第一层和第二层。这第一层(被称作“片段层”)处理 IP 信息包片段。例如，在为出站 IP 信息包分段之前，并在将入站 IP 信息包重新装配成单一 IP 信息包之后，这第二层(被称作“完全装配层”)处理完全的IP信息包。在网络层处理和可能的分裂之后，信息包408(c)被发送到链路层 274。链路层 274 通过为 MAC 头部提供该源 MAC 地址和目的 MAC 地址以及其他信息，来进一步对该数据实行分组(packetizes)。然后，将该信息包发送到网络接口卡(NIC)，在那里，该信息包在物理上被传送到网络上。

用互换方式来处理入站信息包。信息包 408(d)由该 NIC 来接收，并被发送到链路层 274。如果必要的话，则除去该 MAC 头部，并将信息包 408(c)发送到该网络层，在那里，重新装配这些 IP 信息包片段；并且，分析该 IP 头部。然后，该网络层将信息包 408(b)发送到该传输层，在那里，除去该 TCP 头部；并且，如果在多个 TCP 信息包中发送过该数据流，则重新装配该数据流。最后，将数据流 408(a)发送到数据流层 268，在那里，该数据由该应用协议(在这种情况下是该 HTTP 协议)来进行解密。

关于每个出站信息包，该网络堆栈的各个层保持信息包上下文 410(a)-(c)。关于每个入站信息包，该网络堆栈的各个层保持信息包上下文 412(a)-(c)。当每个信息包穿过这些网络层时，该信息包上下文跟随它。信息包上下

文也被传递到呼出 258，并可以由这些呼出来加以修改(图 3)。

当在每个层处处理这些信息包时，更新该信息包上下文。每个层将其层参数加入该信息包上下文，从而将该信息提供给随后的各个层或处理。如所示，链路层 274 加入如上下文 412(a)所展示的源 MAC 地址和目的 MAC 地址以及关于入站信息包的接口号。那个上下文由网络层 272 来接收，网络层 272 加入如上下文 412(b)所展示的源 IP 地址和目的 IP 地址。传输层 266 接收该上下文，并加入如上下文 412(a)所展示的这些端口号。

对于和出站信息包关联的上下文 410(a)-(c)而言，会发生类似的处理。数据流层 268 加入诸如来自如上下文 410(a)所展示的该信息包有效载荷的 URL 地址等信息，传输层 270 进一步加入如上下文 410(b)所展示的源端口号和目的端口号，该网络层加入如上下文 410(c)所展示的源 IP 地址和目的 IP 地址。

将会理解：每个层可以将可用的任何上下文信息加入那个层。通常，这包括该层被设计成要加以处理(即，增加信息包或分析信息包或从信息包中导出)的任何信息。

当在每个层处接收该信息包及其对应的上下文时，通过识别这些层参数并发送被标注为 414 的分类请求，该层可用作请求层。分类请求 414 包括层参数 416、从该先前层接收的信息包上下文 418 和完全信息包 420。被用来发出该分类请求的示范方法是参照层 API 280 来加以描述的“分类”法 350(图 7)。

响应于每个分类请求，核心模式防火墙引擎 256 将层参数 416 和信息包上下文 418 跟被分配给该请求层的这些滤波器的滤波器条件 318(图 4)进行比较。核心防火墙引擎 256 将被标注为 422 的响应(具有来自拥有最高加权 314 的匹配滤波器 310 的动作 424)发送到该请求层。核心防火墙引擎 256 也返回策略上下文 426。如果核心防火墙引擎 256 没有识别匹配滤波器，则该核心防火墙引擎通知该请求层：不存在匹配滤波器。核心防火墙引擎 256 继续识别匹配滤波器，直到匹配滤波器指定终止动作(即允许或阻滞)为止，或者直到检验被分配给该请求层的所有滤波器(不管首先检验哪个滤波器)为止。作为选择，核心防火墙引擎 256 识别所有匹配，并在单一响应中将这些动作返回到该请求层。

如一般所展示的，层参数的识别作为由网络堆栈 254 的各个层执行的该标准层处理的一部分。不要求额外的信息包分析，从而将对系统性能的影响减到最小。而且，由于这些层在保持信息包上下文的过程中进行合作，因此，防火墙引擎 256 可以将滤波器条件与通常无法使用那些信息包参数的各个层处的信

息包参数进行比较。例如，网络层 268 从包括源 MAC 地址和目的 MAC 地址的链路层 274 那里接收关于入站信息包的上下文。由于网络层 272 发出具有网络层参数(例如，源 IP 地址和目的 IP 地址以及该信息包上下文)的分类请求，因此，即使这些 MAC 地址通常不可用，核心防火墙引擎 256 也可以在网络层 272 处的 IP 地址和 MAC 地址上进行过滤。

参考图 9，现在将描述与本发明的该防火墙结构一同包括在内的呼出模块的示范集合 258。呼出模块 258 包括 HTTP 上下文呼出 430、侵入检测呼出 436、IPSec 呼出 438 和记录呼出 440。

HTTP 上下文呼出 430 保持可接受的或(作为选择)不能接受的 URL 地址的高速缓存 432。HTTP 上下文呼出 430 定期访问与该公用网连接的服务器 434，该公用网保持 URL 地址并将它们分类成“可接受的”或“不能接受的”。当核心防火墙引擎 256 执行该 HTTP 上下文呼出时，该呼出检查该信息包，对该 URL 地址进行解密(如果必要的话)，并确定：根据高速缓存 432 中的信息，它是否是可接受的。然后，如果该 URL 地址是可接受的，则该 HTTP 呼出返回允许，作为动作 316；如果该 URL 地址对于核心模式防火墙引擎 256 而言是不能接受的，则返回阻滞；这又经由层 API 280 将动作 316 返回到该请求层。该 HTTP 上下文呼出在执行双亲控制功能的过程中可发挥作用。

侵入检测呼出 436 使用可用的算法和技术来检查该信息包，以识别病毒或可疑信息包的征候。如果检测到可疑信息包，则返回阻滞动作 316。可疑信息包的例子是这样一种信息包—— 其中，IP 头部和 TCP 头部中的所有标记被设置为值 1。该信息包是可疑的，这是因为它永远不会有效，并且会指出攻击签名。作为选择，侵入检测呼出 436 修改信息包上下文，以便标记该信息包的可疑性质，从而推迟了关于是否阻滞到该网络堆栈中的随后各层的该信息包的决定。

IPSec 呼出 438 被设计成确定是否曾将适当的安全协议应用于该信息包。IPSec 呼出 438 与 IPSec 处理进行通信，并根据 IPSec 策略来确定该信息包是否曾被期望经历 IPSec 处理。如果是，则 IPSec 呼出 438 根据信息包上下文来验证：该信息包实际上是否经历过该 IPSec 处理。如果该信息包曾被期望根据 IPSec 来加以处理，但却不是如此(例如，该信息包在明文中)，则返回阻滞动作。如果该信息包经历过 IPSec 处理，则该 IPSec 呼出验证：应用过该适当的 SA。

使用记录呼出 440 来保存关于该信息包(例如，供以后使用的该完全信息包)的信息。例如，这类以后的使用可能是关于某个网络信息流通量(该网络信息流通量出乎意料地没有穿过网络，因为它正受到阻滞)或用于跟踪该系统上的恶意攻击的诊断程序。

图 10 展示了被用来执行本发明的该总体防火墙功能的处理 450。在步骤 452 中，该请求层发出关于信息包的分类请求，从而识别该请求中的信息包参数。在步骤 454 中，识别与该分类请求中的这些信息包参数相匹配的滤波器。然后，根据如步骤 456 中所示的这些匹配滤波器，来决定是否应该丢失该信息包。如果该决定是“丢失该信息包”，则丢失该信息包，并且，该处理结束，而不会执行进一步的信息包处理。如果该决定是“不丢失该信息包”，则该请求层根据在该请求层处所执行的该协议来处理该信息包，并且，如步骤 458 中所示的那样来修改该信息包上下文数据结构。如果没有额外的层，则该处理同样结束。否则，如步骤 462 中所示，将该处理过的信息包和信息包上下文发送到这下一层。该处理继续进行，直到丢失该信息包为止，或者直到该信息包穿过所有层为止。

图 11 展示了被标注为 500 的一种方法，该方法被该请求层用来处理信息包、发出分类请求并保持信息包上下文。这里所描述的该方法由这各种网络层的薄垫片模块 400、402、404、406 来执行。作为选择，方法 500 由来自该网络堆栈的这些层内的整体处理来加以执行，而不需要单独的薄垫片。方法 500 也可以由一个或多个用户模式层来执行。

在步骤 502 中，该请求层从前一层接收完全信息包 408 和对应的信息包上下文 412。在出站信息包的情况下，这前一层在该网络堆栈中比该请求层更高。在入站信息包的情况下，这前一层在该网络堆栈中比该请求层更低。

在步骤 504 中，该请求层识别这些层参数。这些层参数由该请求层通过从入站信息包中分析这些参数或将这些参数加入出站信息包来加以处理。这些层参数可以包括可从信息包中导出的其他信息(例如，该局部地址类型)。以上的表格 A 中描述了这些默认层参数，并且，层 API 280 中的“分类”法 350 的该 pInFixValues 中包括这些默认层参数。

在步骤 506 中，该请求层将该分类请求发给核心防火墙引擎 256。参照层 API 280 中的“分类”法 350 来描述过用于发出分类请求的示范方法。

响应于该分类请求，将动作返回到该请求层。该请求层根据被返回的这个

动作来确定是否丢失信息包 508。如果该核心防火墙引擎返回阻滞，作为该动作，那么，该请求层丢失该信息包。如果核心防火墙引擎 256 返回该动作，因为没有发现匹配滤波器，那么，该请求层也可以丢失该信息包。“该请求层是否在没有发现匹配滤波器的情况下丢失信息包”这一点在系统范围的基础上或在逐层的基础上是可以配置的。

如果被返回的这个动作是允许，则进一步进行信息包处理。在步骤 510 中，该请求层将该信息包上下文修改成包括层信息，通常是曾作为该分类请求中的这些层参数而被包括在内的相同类型的信息。这样，表格 A(以上)不仅识别这些层参数，而且识别由每个层加入该信息包上下文的该默认信息。该信息包上下文被保存在数据结构(例如，参照图 5 而描述的那个数据结构)中。

在步骤 512 中，该请求层根据关于那个层的该协议实施来处理该信息包。这类处理众所周知，这里不需要详细描述。示范的协议实施包括 HTTP、FTP、SMTP、关于该应用层的 RPC、关于该传输层的 TCP 和 UDP、关于该网络层的 IP，以及关于该链路层的 NDIS。

在步骤 514 中，该请求层将该信息包(根据该层协议来加以处理)连同该修改过的信息包上下文传递到下一层。如果该信息包是入站信息包，则这下一层是该网络堆栈中的较高的层。如果该信息包是出站信息包，则这下一层是该网络堆栈中的较低的层。处理 500 由该网络堆栈中的每个层来重复执行，并且继续进行，直到该信息包穿过该网络堆栈中的所有层为止，或者直到该信息包被该网络堆栈中的这些层之一丢失为止。

参考图 12，现在将描述示范方法 520，示范方法 520 被核心防火墙引擎 256 用来识别匹配滤波器并将该动作返回到该请求层。如前所述，也可以由在该用户防火墙引擎中执行的滤波器模块 294 来实现核心防火墙引擎 256 的功能性。

在步骤 522 中，核心防火墙引擎 256 使用(例如)“分类”法 350 并经由层 API 280，从该请求层接收该信息包、层参数和信息包上下文。

在步骤 524 中，该核心防火墙引擎识别一个或多个匹配滤波器。在本发明的实施例中，将滤波器分配给特定的请求层。如果将该滤波器分配给该请求层，并且，这些信息包参数与所有的滤波器条件 318 相匹配，则核心防火墙引擎 256 只尝试将滤波器识别为匹配(图 5)。如前所述，这些信息包参数包括来自这些请求层的层参数和该信息包上下文。在识别所有的匹配滤波器之后，该核心防火墙引擎根据每个匹配滤波器中的加权字段 314 来安排这些滤波器。

在步骤 526 中，核心防火墙引擎 256 应用具有还未被应用的最高加权 314 的那个滤波器。明确地说，核心防火墙引擎 256 识别该滤波器中所规定的动作 316。如果动作 316 指定呼出模块 258 之一，则核心防火墙引擎 256 经由如步骤 532 中所示的呼出 API 284 来执行该呼出。参照该呼出 API 中的“分类”法 356(图 6)来描述过被用来执行该呼出的示范方法。该呼出可以将动作返回到核心防火墙引擎 256。

如果该动作不是呼出，或者在执行该呼出之后，该核心防火墙引擎将来自该匹配滤波器的或如该呼出所识别的关联动作返回到如步骤 536 中所示的该请求层。作为选择，核心防火墙引擎 256 等候返回该动作，直到已应用所有的匹配滤波器为止。

在步骤 534 中，核心防火墙引擎 256 确定是否存在任何额外的匹配滤波器。如果不存在，则该处理终止。如果存在额外的匹配滤波器，则该处理返回到步骤 526，在那里，应用优先级高居第二的滤波器。该处理继续进行，直到应用所有的匹配滤波器为止。作为选择，一旦为该信息包识别终止动作，该处理就终止。如果没有为该信息包识别匹配滤波器，则该核心防火墙引擎通知该请求层：不曾发现匹配滤波器。然后，该请求层确定如何处置该信息包(即，是允许还是阻滞该信息包)。

图 13 展示了根据本发明而使用的方法 560，用于防止来自未经请求的入站信息包的恶意攻击，同时允许与可信用户进行通信，从而启动来自未知网络地址的通信。

在步骤 562 中，执行本发明的该防火墙的响应计算机接收未经请求的入站信息包。在步骤 564 中，处理 560 确定该入站信息包是否是鉴定请求。例如，鉴定请求可能取决于该 IKE 协议。如果该入站信息包是鉴定请求，则该响应计算机尝试鉴定如步骤 568 中所示的该启动计算机。被用来鉴定该启动计算机的一种方法是通过数字证书。该数字证书由认证当局(CA)发行，它包括诸如用户名、序列号、截止日期、公开密钥(被用于为消息和数字签名加密)的副本和该 CA 的数字签名等信息，以便接收者可以验证：该证书是真实的。

然后，在步骤 568 中，该处理确定该鉴定处理是否成功。如果该鉴定不成功(即，该响应计算机无法鉴定该启动计算机)，则该处理结束。

如果该鉴定处理是成功的，则如步骤 570 中所示的那样来创建新的滤波器。这个新的滤波器包括滤波器条件，这些滤波器条件与关于该响应计算机的地址

信息(例如，IP 地址、端口号、协议类型和类似的信息)相匹配；并且作为该关联的动作“允许”而包括在内。作为选择，这个新的滤波器具有滤波器条件，这些滤波器条件跟在该启动计算机与响应计算机之间达成一致意见的安全设置相匹配。如步骤 572 中所示，当接收不是鉴定请求的入站信息包时，该处理确定是否已鉴定该启动计算机(即，是否存在具有允许动作的匹配滤波器)。如果匹配滤波器具有该允许动作，则如步骤 547 中所示的那样，允许该信息包穿过网络。否则，如步骤 576 中所示的那样，阻滞该信息包进行进一步的网络遍历。

用于提供来自未经请求的用户的安全通信的方法 516 的替换方案是：创建一种滤波器，该滤波器要求所有的入站信息包符合安全协议(例如，由 IPSec 规定的协议)。这样，在信息包可以畅通无阻地通过该滤波器之前，它必须如步骤 566 中所描述的那样，首先建立被鉴定的 SA。

这里所引用的所有这些参考资料被包括于此，用作整体参考。

鉴于可以应用本发明的这些原理的这许多可能的实施例，应该认识到：这里根据这些附图而描述的该实施例意在只起说明的作用，而不应该被视作限制本发明的范围。例如，精通该技术领域的人将会认识到：可以在硬件中执行软件中所示的该说明性实施例的这些元件，反之亦然；或者，在不脱离本发明的精神的前提下，可以在布置和细节方面修改该说明性实施例。此外，精通该技术领域的人将会认识到：其他处理使用滤波器(例如，QOS 和 IPSec)。可以使用本发明的这项发明来管理并执行滤波器以及这些和额外的处理的基于滤波器的策略。所以，如这里所描述的本发明计划可以将所有这类实施例纳入以下的权利要求书及其相等物的范围以内。

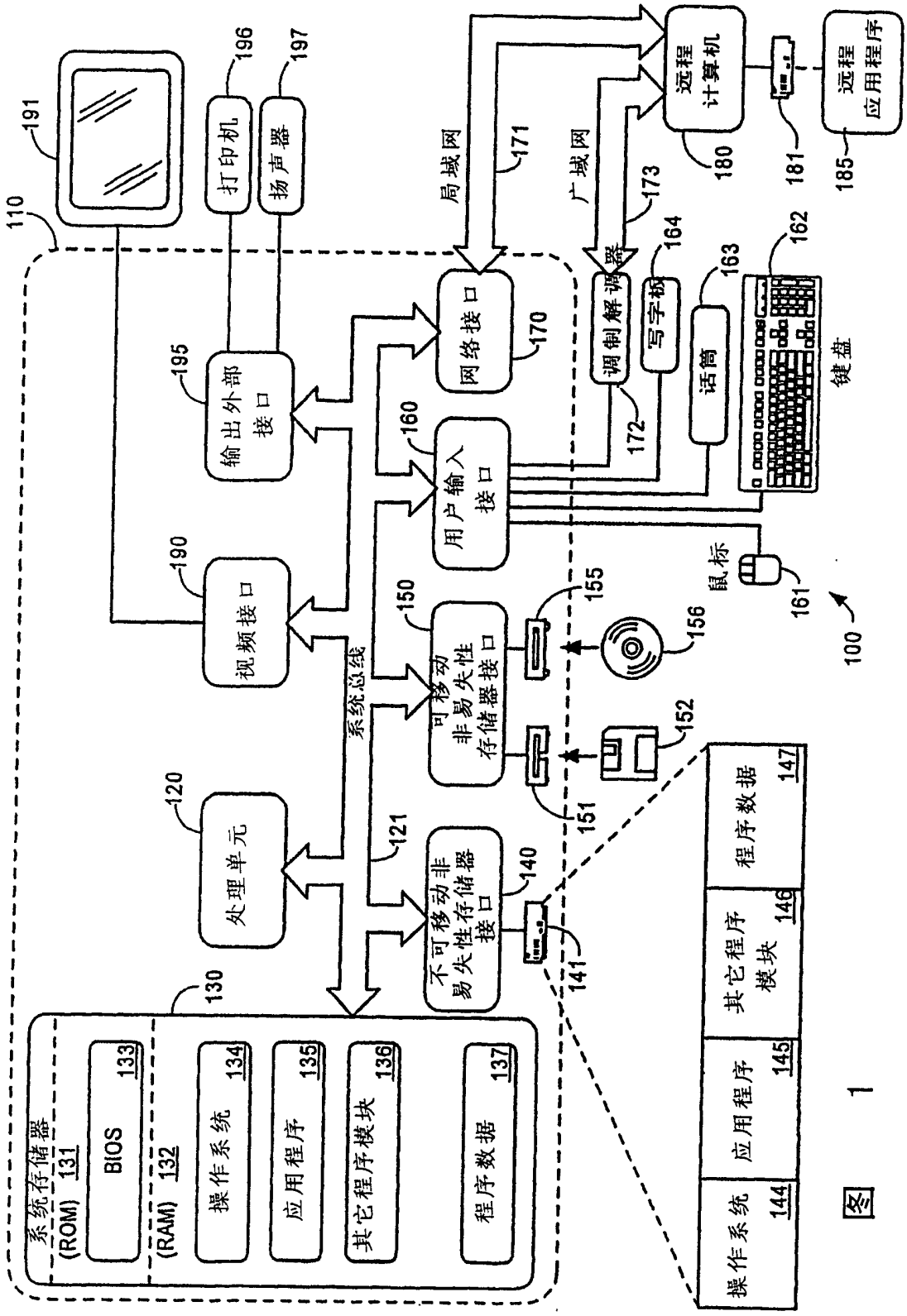

110
191
196
打印机
197
扬声器
局域网
171
广域网
173
远程计算机
180
181
185
远程应用程序
195
输出外部接口
网络接口
170
调制解调器
172
164
写字板
163
话筒
162
键盘
160
用户输入接口
鼠标
161
100
190
视频接口
150
可移动非易失性存储器接口
155
156
系统总线
120
处理单元
121
152
151
140
不可移动非易失性存储器接口
141
130
系统存储器
(ROM) 131
BIOS 133
(RAM) 132
操作系统 134
应用程序 135
其它程序模块 136
程序数据 137
操作系统 144
应用程序 145
其它程序模块 146
程序数据 147

图 1

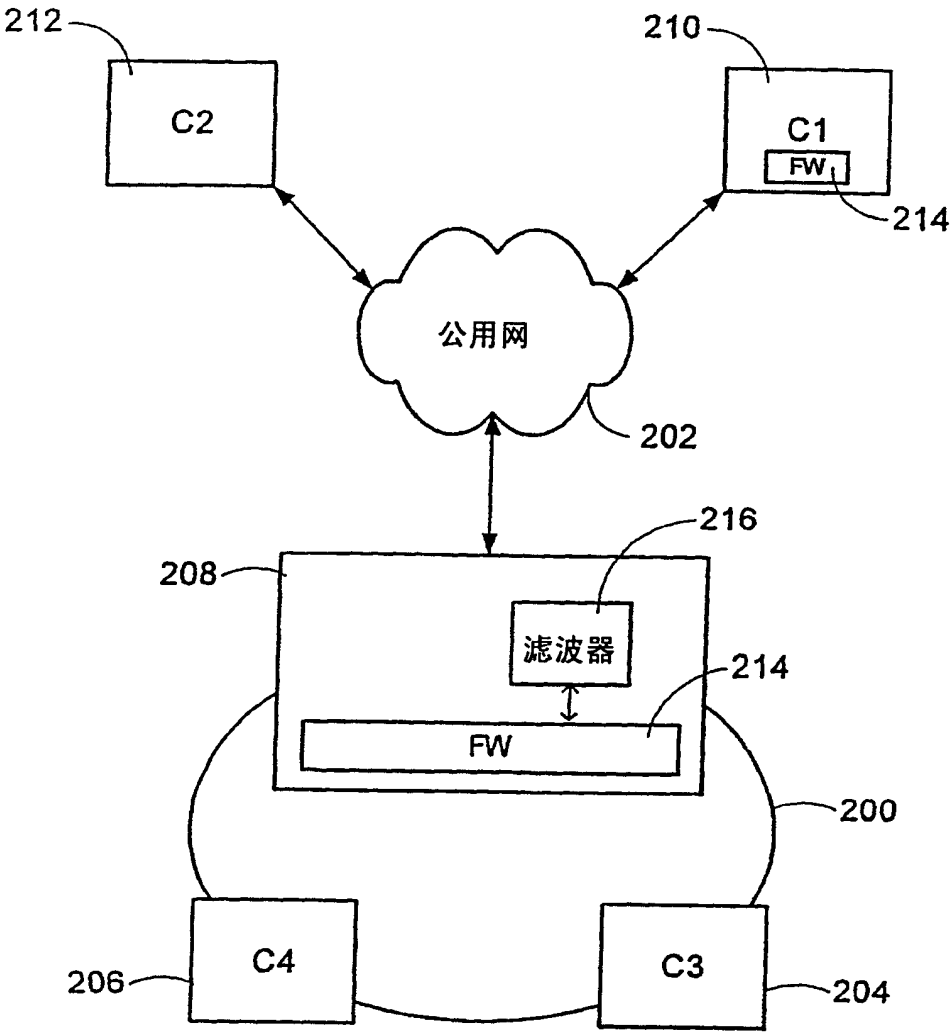


图 2

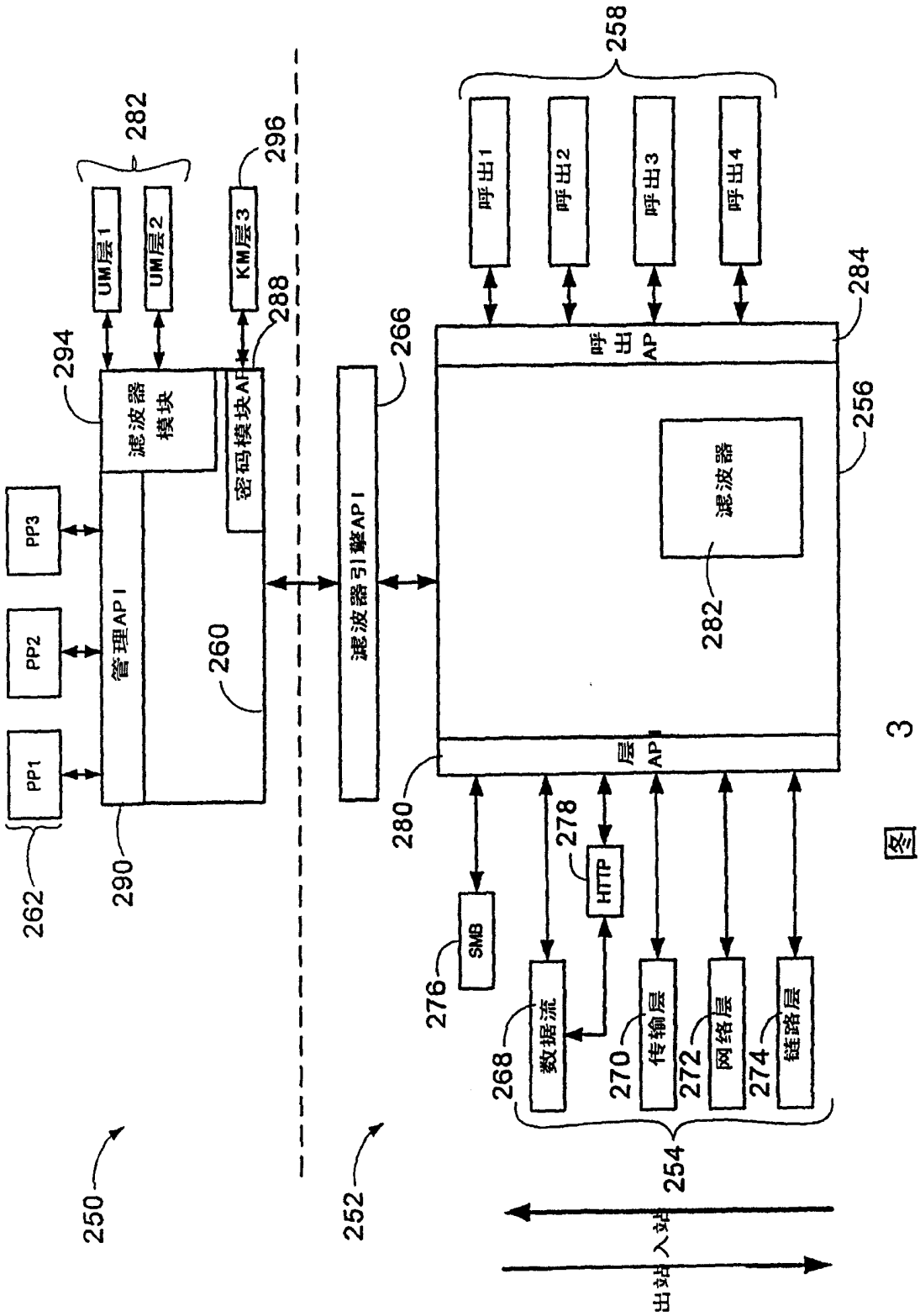

250
252
254
256
258
260
262
266
268
270
272
274
276
278
280
282
284
288
290
294
296
PP1
PP2
PP3
管理API
滤波器模块
密码模块API
UM层1
UM层2
KM层3
滤波器引擎API
呼出1
呼出2
呼出3
呼出4
呼出AP
滤波器
层API
SMB
HTTP
数据流
传输层
网络层
链路层
入站
出站

图 3

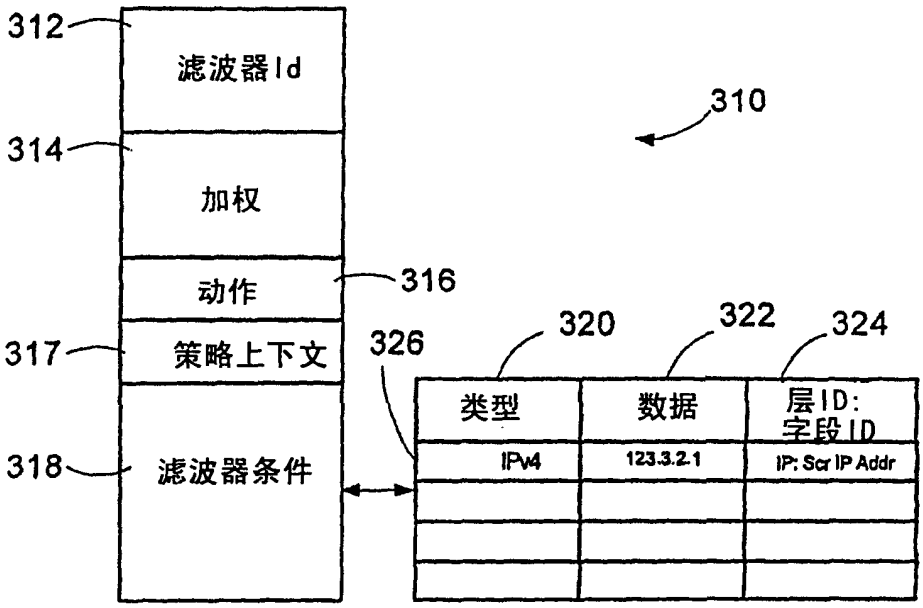

312
滤波器Id
314
加权
动作
316
317
策略上下文
318
滤波器条件
310
320
322
324
326
类型
数据
层ID:
字段ID
IPv4
123.3.2.1
IP: Scr IP Addr

图 4

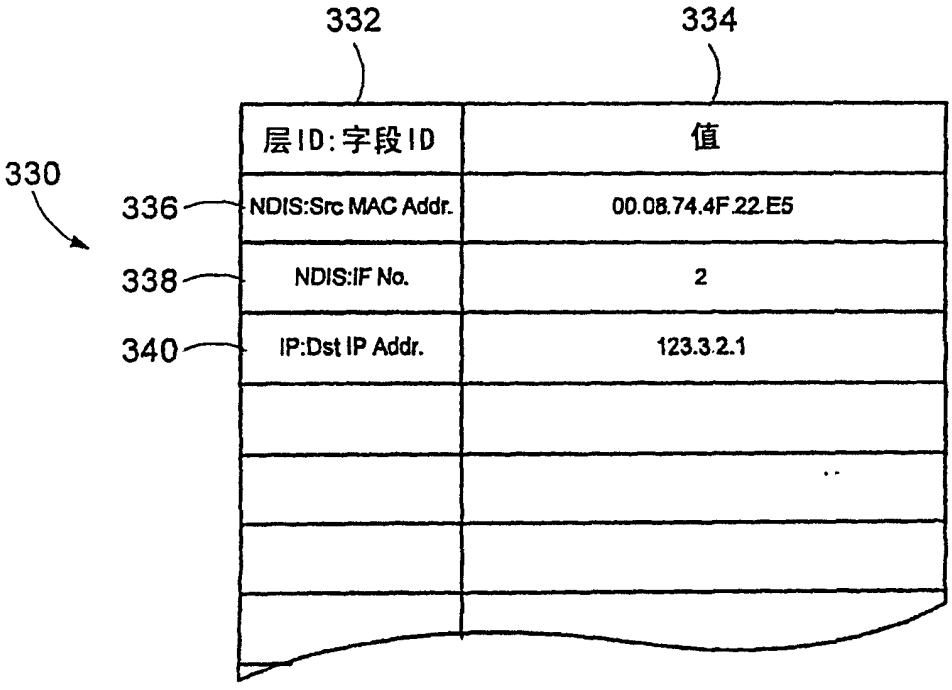

332
334
330
层ID:字段ID
值
336
NDIS:Src MAC Addr.
00.08.74.4F.22.E5
338
NDIS:IF No.
2
340
IP:Dst IP Addr.
123.3.2.1

图 5

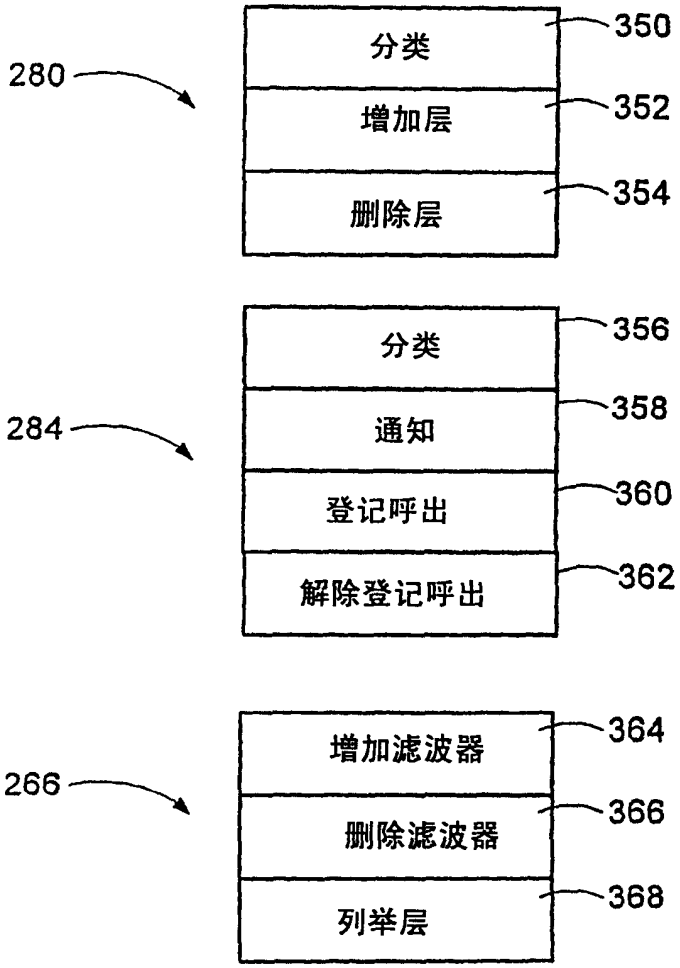

280
分类
350
增加层
352
删除层
354
284
分类
356
通知
358
登记呼出
360
解除登记呼出
362
266
增加滤波器
364
删除滤波器
366
列举层
368

图 6

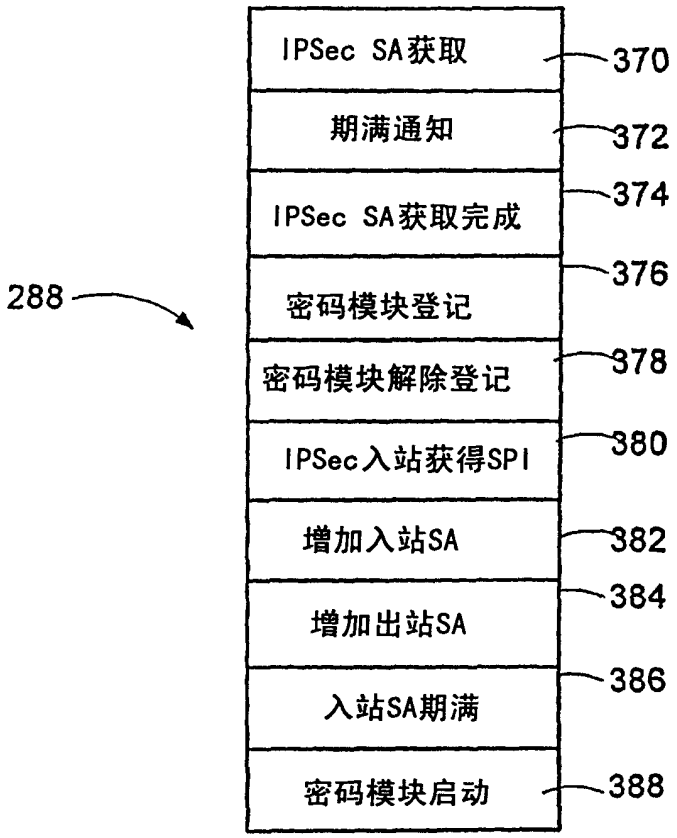

IPSec SA获取
370
期满通知
372
IPSec SA获取完成
374
密码模块登记
376
288
密码模块解除登记
378
IPSec入站获得SPI
380
增加入站SA
382
增加出站SA
384
入站SA期满
386
密码模块启动
388

图 7

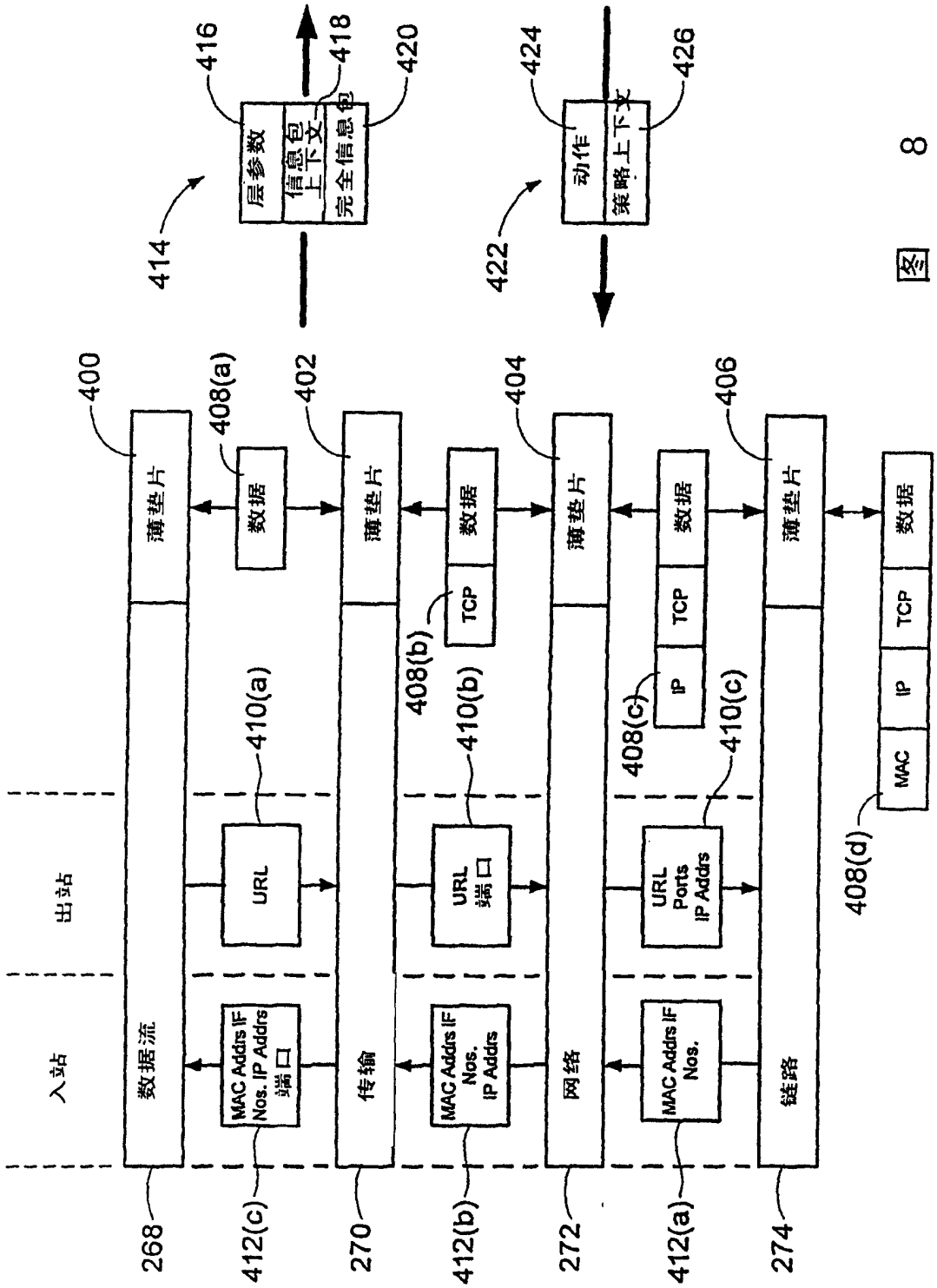

入站
出站
数据流
薄垫片
MAC Addrs IF Nos. IP Addrs 端口
URL
数据
传输
薄垫片
MAC Addrs IF Nos. IP Addrs
URL 端口
TCP
数据
网络
薄垫片
MAC Addrs IF Nos.
URL Ports IP Addrs
IP
TCP
数据
链路
薄垫片
MAC
IP
TCP
数据
层参数
信息包上下文
完全信息包
动作
策略上下文
400
402
404
406
408(a)
408(b)
408(c)
408(d)
410(a)
410(b)
410(c)
412(a)
412(b)
412(c)
414
416
418
420
422
424
426
268
270
272
274

图 8

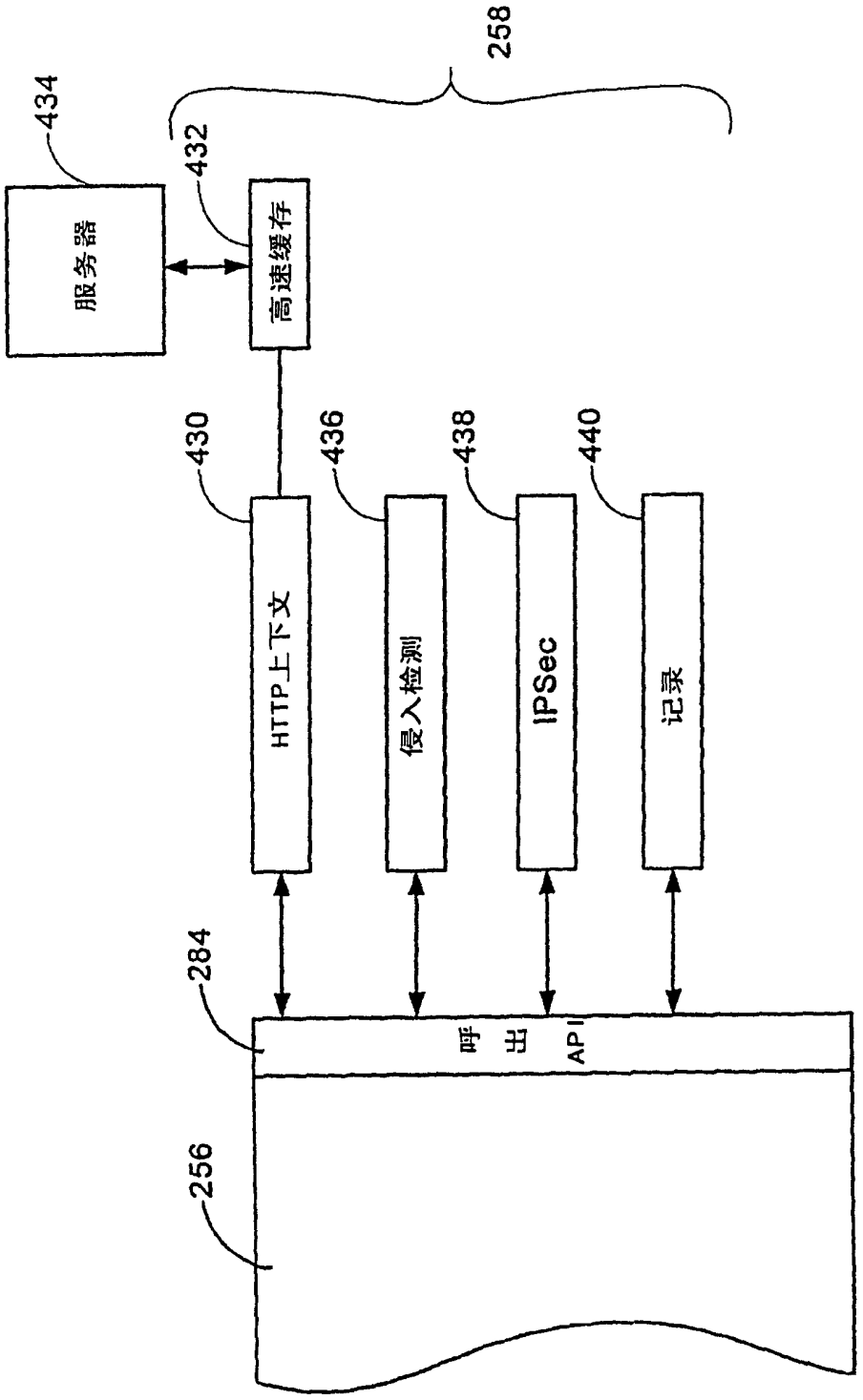

258
434
432
服务器
高速缓存
430
436
438
440
HTTP上下文
侵入检测
IPSec
记录
284
呼 出 API
256

图 9

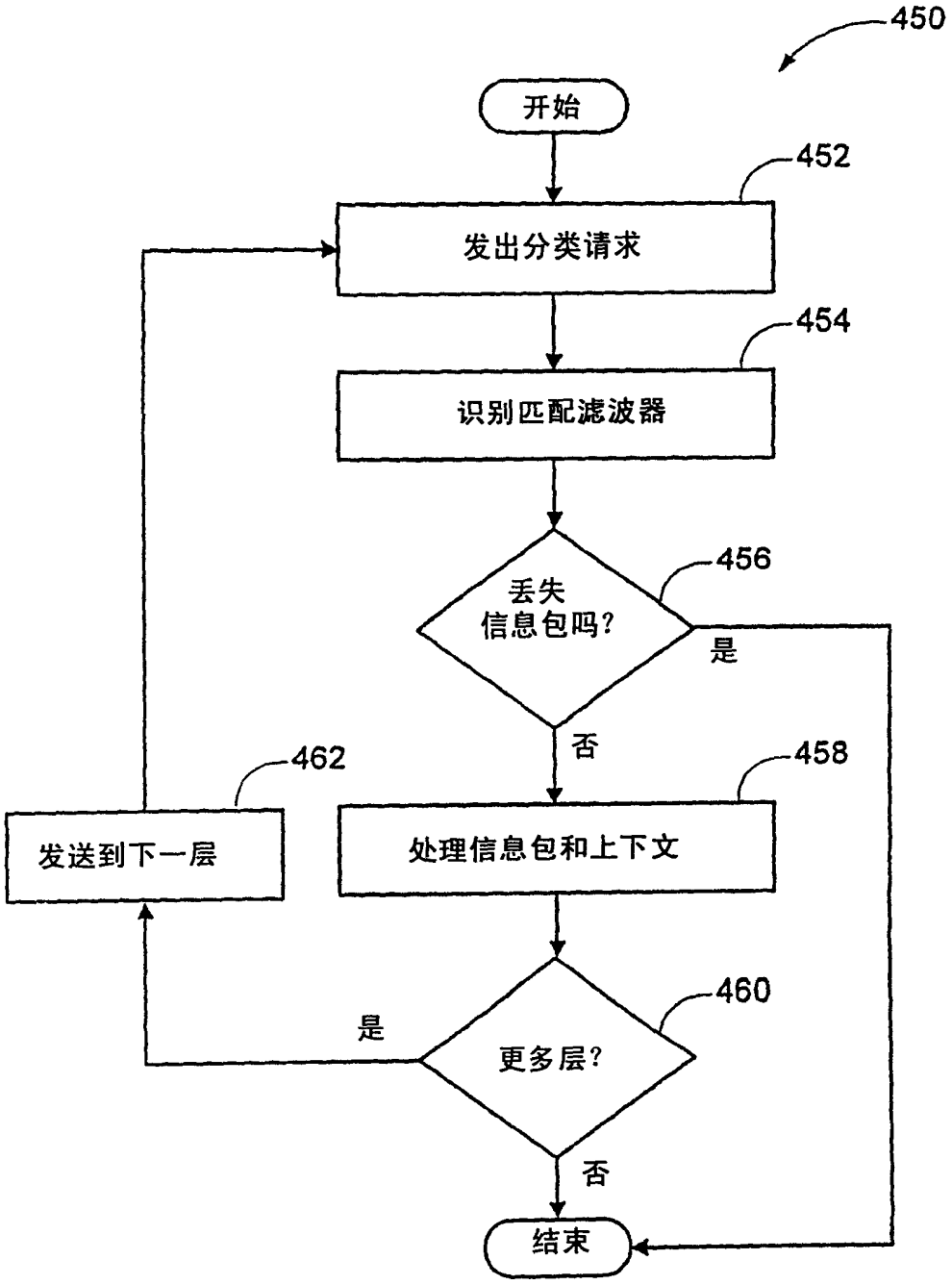

450
开始
452
发出分类请求
454
识别匹配滤波器
456
丢失
信息包吗?
是
否
462
发送到下一层
458
处理信息包和上下文
460
是
更多层?
否
结束

图 10

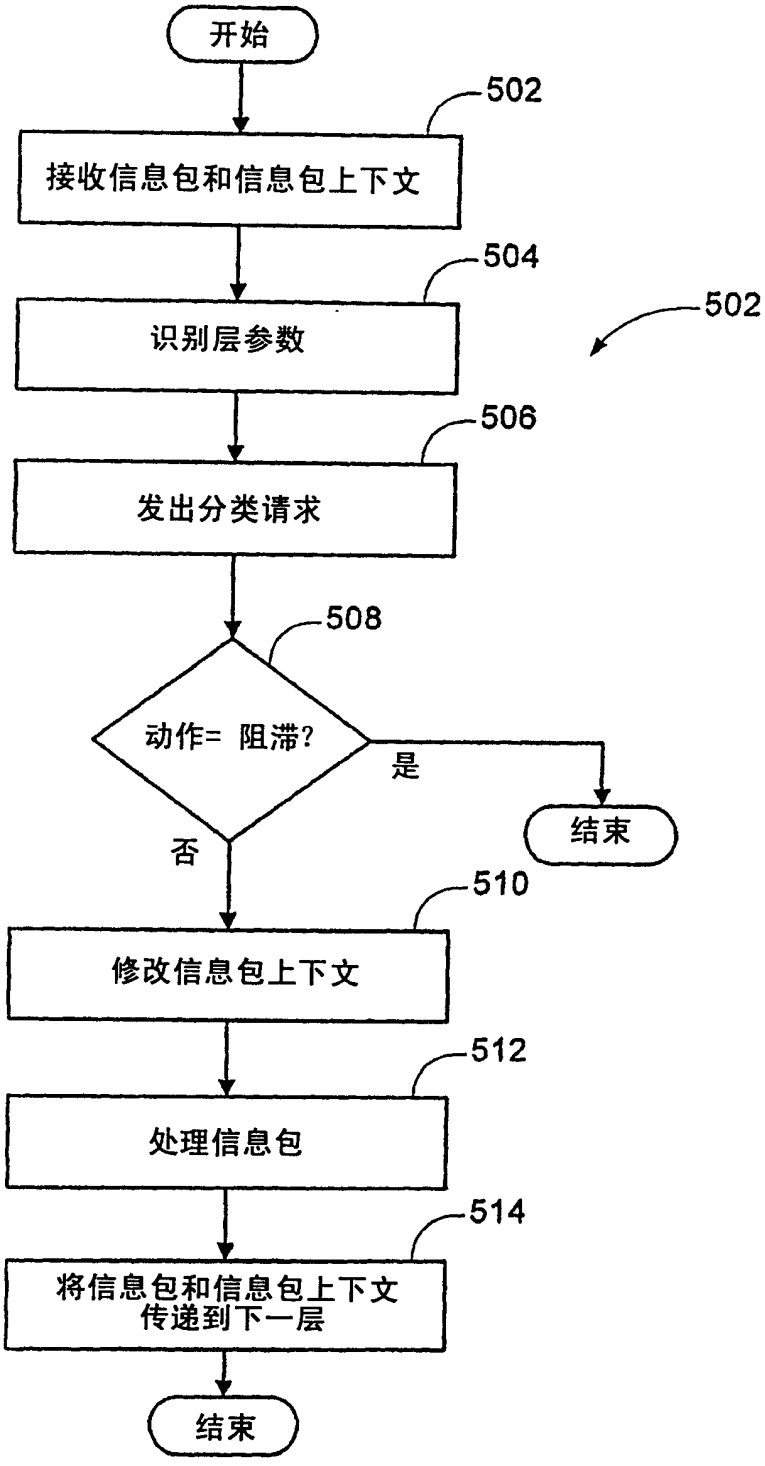

开始
502
接收信息包和信息包上下文
504
识别层参数
502
506
发出分类请求
508
动作= 阻滞?
是
结束
否
510
修改信息包上下文
512
处理信息包
514
将信息包和信息包上下文
传递到下一层
结束

图 11

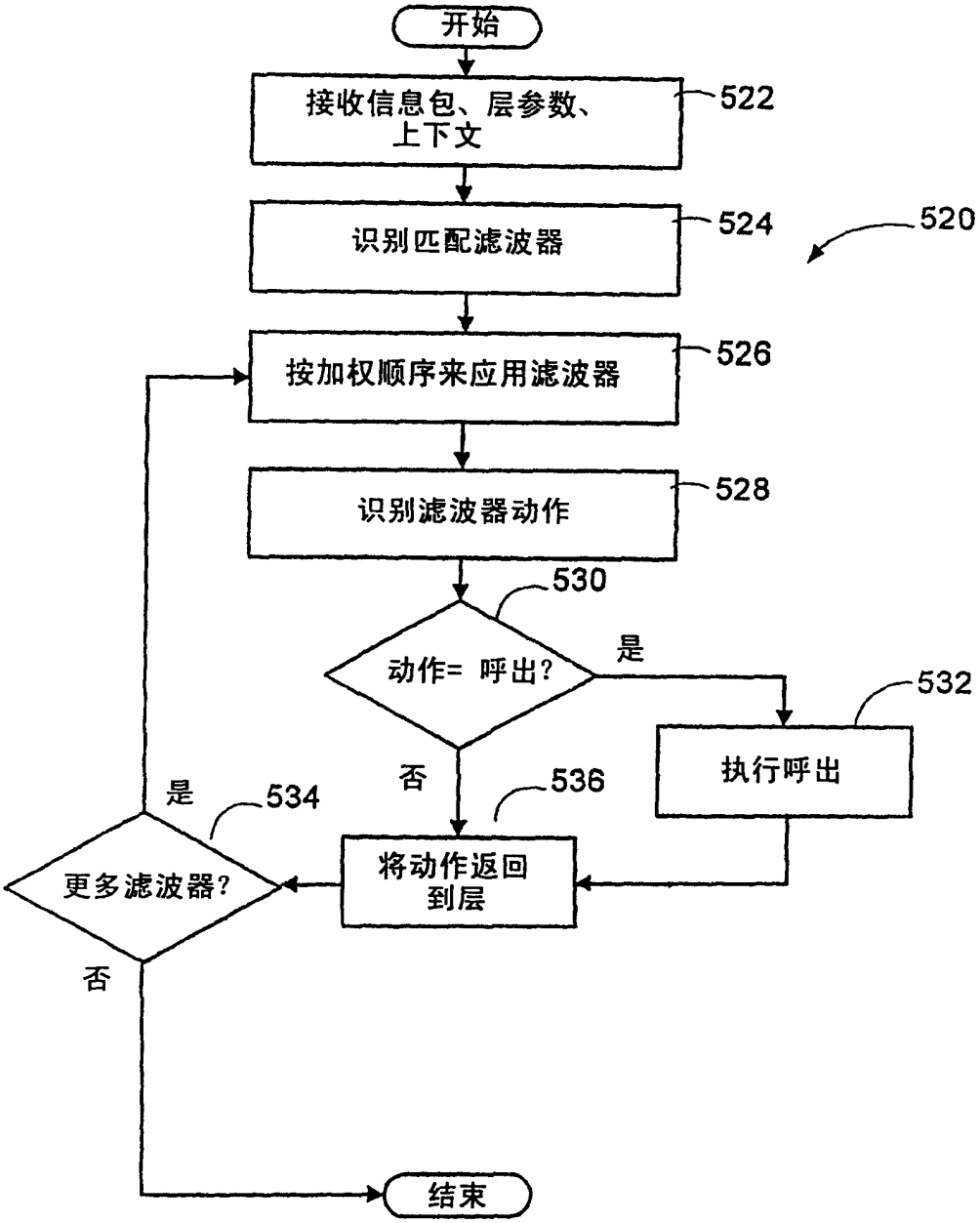

开始
接收信息包、层参数、上下文
522
识别匹配滤波器
524
520
按加权顺序来应用滤波器
526
识别滤波器动作
528
530
动作= 呼出?
是
532
执行呼出
否
536
将动作返回到层
是
534
更多滤波器?
否
结束

图 12

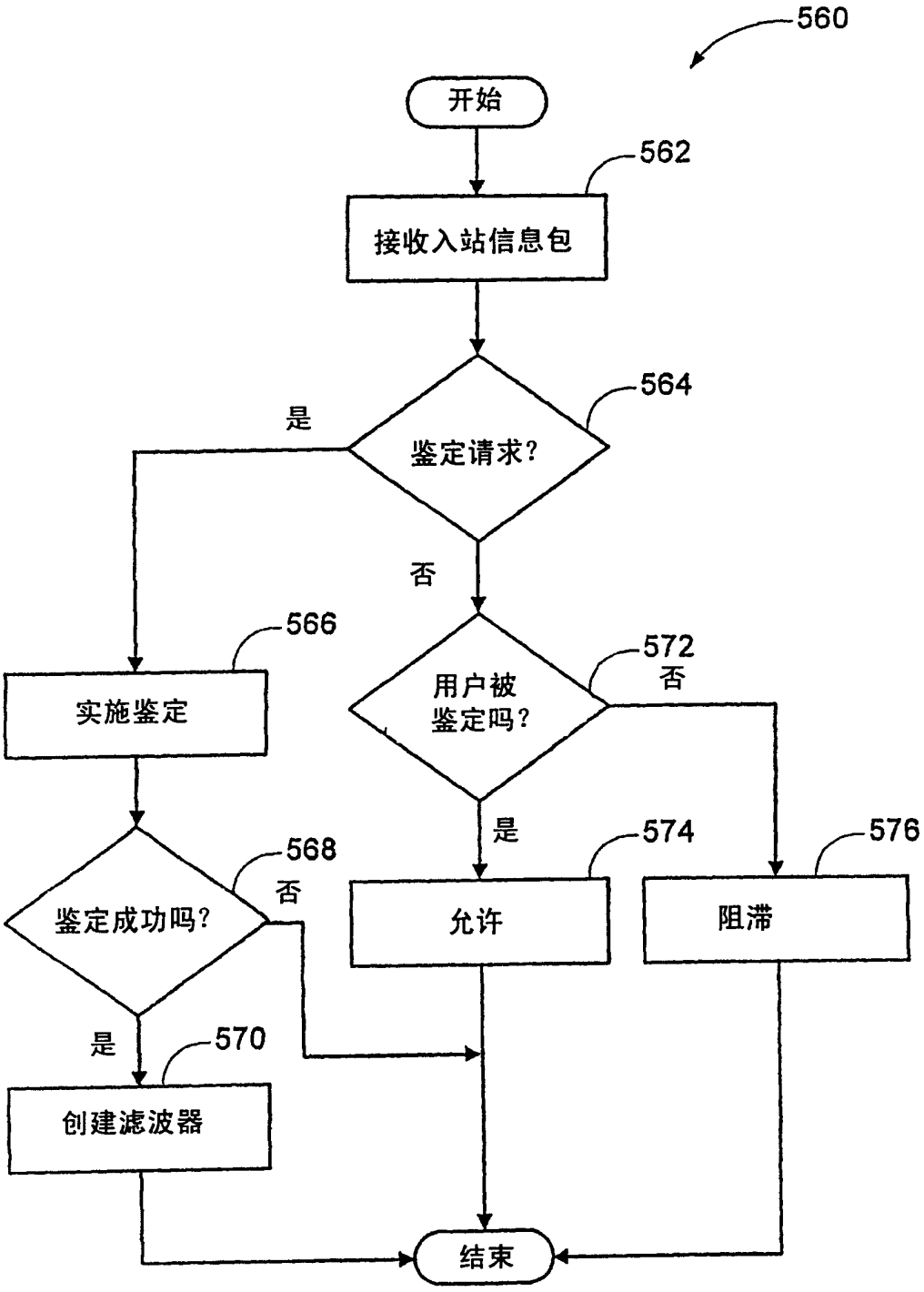

560
开始
562
接收入站信息包
564
鉴定请求?
是
否
566
实施鉴定
572
用户被鉴定吗?
否
是
568
鉴定成功吗?
否
是
570
创建滤波器
574
允许
576
阻滞
结束

图 13