发明名称
使用最具体的过滤器匹配和传输层共享进行两级分组分类的方法和装置

摘要
用于两级分组分类的方法和装置，所述两级分组分类方案包括第一级和第二级。在第一分类级中，基于分组的网络路径对分组进行分类。在第二分类级中，基于所述分组的一个或更多个传输（或其他）标识对分组进行分类。还公开了最具体的过滤器匹配和传输层共享的实施方案，并且这些技术中的任意一种或两种都可以在所述两级分类方法中实现。
1. 一种用于对分组进行分类的方法，包括：
   接收分组，所述分组具有包括源地址、目的地址和多个其他字段的头部；
   从数据结构中的多个项中识别具有与所述分组的所述源地址匹配的源地址前缀的项，
   所述匹配项包括第一标识符；
   从另一个数据结构中的多个项中识别具有与所述分组的目的地址匹配的目的地址
   前缀的项，所述匹配项包括第二标识符；
   从多个容器中识别与所述第一和第二标识符对应的容器。所述对应的容器包括多个传输
   层字段集，以及
   将所述分组头部的其他字段中的至少一个与所述对应的容器中的每一个传输层字段
   集进行比较，以寻找到匹配的传输层字段集。

2. 如权利要求 1 所述的方法，还包括将与所述匹配传输层字段集相关联的动作应用于
   所述接收的分组。

3. 如权利要求 1 所述的方法，其中，所述分组头部中的所述多个其他字段包括协议、
   源端口以及目的端口中至少一个。

4. 如权利要求 3 所述的方法，其中，所述分组头部的所述源地址包括源 IP（因特网
   协议）地址，并且所述分组头部的所述目的地址包括目的 IP 地址。

5. 一种用于对分组进行分类的方法，包括：
   接收分组，所述分组具有包括源地址、目的地址和多个传输层字段的头部；
   搜索源地址数据结构，以寻找到第一索引和第三索引，所述第一索引与完全指明的过
   滤器相关联，所述完全指明的过滤器具有与所述分组的所述源地址匹配的源前缀，所述第
   三索引与部分指明的过滤器相关联，所述部分指明的过滤器具有与所述分组的所述源地址
   匹配的源前缀；
   搜索目的地址数据结构，以寻找到第二索引和第四索引，所述第二索引与完全指明的
   过滤器相关联，所述完全指明的过滤器具有与所述分组的所述目的地址匹配的目的前缀，
   所述第四索引与部分指明的过滤器相关联，所述部分指明的过滤器具有与所述分组的所述
   目的地地址匹配的目的前缀；
   从所述第一和第二索引形成关键字；
   搜索主表以寻找与所述关键字匹配的项，所述主表包括多个项，每个项与完全指明的
   过滤器、完全指明的过滤器的交叉以及指示器过滤器中的一个对应；
   如果在所述主表中寻找到匹配项，则访问与所述匹配项相关联的容器指针的列表，所
   述列表的每个容器指针标识包括多个传输层字段集的容器；
   访问由所述主表的所述匹配项中的所述容器指针中的一个标识的容器中的一个；
   将所述分组的所述传输层字段与所述被访问容器中的每个传输层字段集进行比较；以
及

如果所述被访问的容器具有与所述分组的所述传输层字段匹配的传输层字段集，则将与所述匹配传输层字段集相关联的动作应用于所述接收的分组。

6．如权利要求 5 所述的方法，还包括：
搜索两个次表中的第一个以寻找与所述第三索引匹配的项，所述第一次表包括多个项，每个项与所述索引的过滤器对应；
搜索两个次表中的第二个以寻找与所述第四索引匹配的项，所述第二次表包括多个项，每个项与所述索引的过滤器对应；以及
如果在所述主表中没有寻找到匹配，并且在所述两个次表中的一个中寻找到匹配项，则访问与所述匹配项相关联的容器指针列表，所述列表的每个容器指针标识包括多个传输层字段集的容器。

7．如权利要求 6 所述的方法，还包括：
访问由一个次表的所述匹配项中的所述容器指针中的一个标识的容器中的一个；
将所述分组的所述传输层字段与所述被访问容器中的每个传输层字段进行比较；以及
如果所述被访问的容器具有与所述分组的所述传输层字段匹配的传输层字段集，则将与所述匹配的传输层字段集相关联的动作应用于所述接收的分组。

8．如权利要求 6 所述的方法，还包括：
如果在所述次表中的任一个中都没有寻找到匹配，则访问与默认项相关联的容器指针列表，所述列表的每个容器指针标识包括多个传输层字段集的容器。

9．如权利要求 8 所述的方法，其中，所述默认项与整个二维地址空间对应。

10．如权利要求 5 所述的方法，还包括：
搜索所述源地址数据结构，以寻找到与宽过滤器相关联的第五索引，所述宽过滤器具有与所述分组的所述源地址数据结构匹配的源前缀。
搜索所述目的地址数据结构，以寻找到与宽过滤器相关联的第六索引，所述宽过滤器具有与所述分组的所述目的地址匹配的目的前缀。
从所述第五和第六索引形成第二关键字；
搜索宽过滤器表以寻找与所述第二关键字匹配的项，所述宽过滤器表包括多个项，每个项与宽过滤器对应；以及
如果在所述主表中没有寻找到匹配，并且在所述宽过滤器表中寻找到匹配项，则访问与所述宽过滤器表中的所述匹配项相关联的容器指针的列表，所述列表的每个容器指针标识包括多个传输层字段集的容器。

11．如权利要求 10 所述的方法，还包括：
访问由所述宽过滤器表的所述匹配项中的所述容器指针中的一个标识的容器中的一个；
将所述分组的所述传输层字段与所述被访问容器中的每个传输层字段集进行比较；以及
如果所述被访问的容器具有与所述分组的所述传输层字段匹配的传输层字段集，则将与所述匹配传输层字段集相关联的动作应用于所述接收的分组。

12. 如权利要求 10 所述的方法，其中，包括在所述宽过滤器表中的每个宽过滤器包括具有多个超过指定阈值的指示器过滤器的完全指明的过滤器。

13. 如权利要求 5 所述的方法，其中，在所述接收的分组中的所述多个传输层字段包括源端口、目的端口以及协议中的至少一个。
使用最具体的过滤器匹配和传输层共享进行两级分组分类的方法和装置

发明领域

本发明总体上涉及计算机网络，并且更具体地，涉及分类分组的方法和装置。

发明背景

传统地，在计算机网络中的分组（packet）路由（route）仅仅基于分组该分组的目的地址。这种路由技术一般与“最好结果”传递相关联，并且去往相同地址的所有流量被相同地对待。然而，单单基于目的地址的分组路由不足以满足对于更宽泛的带宽提升的安全性以及增加的灵活性和服务区分（differentiation）的增长的需求。为了达到这些目标，装备供应商和服务提供商正在提供更加灵活的路由模式，包括通过防火墙（firewall）的路由、基于服务质量（QoS）的转发（forward），以及带宽和/或资源预留（reservation）。

一般来说，防火墙包括任何能够阻挡某些流量类别（例如“不需要的”或“可疑的”流量）的组件或组件的组合。防火墙常常被用在公司网络或其他企业网络中，并且防火墙通常被实施在网络的入口点（entry point）和/或出口点（exit point）——即网络的“信任边界（trust boundary）”。典型的防火墙包括一系列被设计来执行期望的安全策略的规则（rule）或过滤器（filter）。

网络服务提供商可能具有广泛的客户群，各要求不同的服务、服务优先级和价格。为了向大量不同的客户提供不同的服务——或者更一般地，为了向某些网络流量类别提供优惠（preferential）的对待——装备供应商已经实现了多种机制，包括基于QoS的转发以及带宽和/或资源预留。基于QoS的转发的目标在于为大量不同的客户和/或流量类型提供服务区分。基于QoS的转发可以包括例如基于服务类别的转发、特定的排队程序（例如每流排队）和公平调度方法。与QoS转发固有关联的是带宽或资源预留。带宽预留一般包括为某些类型的流量预留指定的带宽。例如，带宽预留可以应用到两点之间的流量，或者带宽预留可以应用到与某种应用（例如多媒体、视频等）相关的流量上。

为了实现上面描述的提供更多各异的网络流量路由的路由方法论（例如防火墙、QoS转发、带宽预留），以及为了执行其他基于策略的分组转发技术，必须要对分组进行分类。一般来说，分组分类包括在属于不同流（flow）的分组之间，或是在于不同流量类别相关联的分组之间进行区分。使用在这里，“流”是一连串分组，这些分组共享至少一些公共的头部（header）特性（例如，在两个具体地址间流动的分组）。通常基于分组头部中的一个或更多个字段（field）对分组进行分类。向该头部信息应用一个或更多个规则来确定该分组与哪个流对应，或是该分组与哪种流量类型相关联。
分组分类规则一般包括几个字段以及相关联的优先级和动作（action），所述字段与接收的分组（received packet）的头部中对应字段进行比较。组成分类数据库的规则组可以被安排到一个按优先顺序的列表中，并且具有较高优先级的规则在其具有较低优先级的规则之上被优选。当接收到分组时，将分组的内容（例如某些头部字段）与分类数据库中的每一条规则进行比较，以确定要应用于该分组的最高优先级的动作。

基于头部数据来执行分组分类的多种方法——硬件实现和软件实现——是本领域公知的，包括哈希（hashing）方案、位并行性（bit parallelism）技术和使用内容可寻址存储器（CAM）的实现。哈希方法根据在规则的每个字段中使用的位掩码（bit mask）创建规则组，每个规则组由一个哈希表（或多个哈希表）代表。识别与接收的分组匹配的规则需要在哈希表上进行一系列的查找。

位并行性把 n 维分类问题分割成多个级，每个级为单维。每次在单维上的匹配都返回一个位向量。位向量具有与存储在系统中的规则的数目相等的长度，并且如果与一个位对应的规则指明值的范围匹配接收的分组的适当字段，则位向量中的该位被置位。使得在所有返回的位向量中的某规则的位被置位的规则匹配所接收的分组。对标准位并行性方案的一种改进是聚合位向量（ABV）方法。对于 ABV 方法来说，每个 “完整” 位向量被压缩，并且被表示为大小更小的位集（被称作 “聚合位向量”）。聚合位向量中的每个位表示来自完整位向量的一个位组，并且如果位组（在完整位向量中）中至少一个位被置位，则聚合位向量中相关联的位被置位。

对于 CAM 实现来说，CAM 的每一项都与值以及位掩码相关联。所述值包括规则的一个或更多个字段，并且所述位掩码指明当搜索关键字（key）与所述值进行比较时所述关键字的哪些位被考虑。CAM 单元——所述 CAM 单元能够同时将搜索关键字与多项进行比较——返回与最高优先级匹配项相关联的索引，并且该索引被用于识别对分组的动作。

多种因素可能影响上述分类方案的性能，所述因素包括需要大量存储器访问、需要大的储存空间以及（至少对于 CAM 实现来说）大量功率消耗。由于带宽和存储器开销（overhead）以及其他因素，这些分组分类技术可能难以与链路速度进步以及分类数据库大小增长保持一致，并且分组分类可能是支持高速链路（例如吉比特容量）的路由器（router）的瓶颈。

附图说明

图 1 是示出包括路由器的网络的实施方案的示意图。
图 2 是示出图 1 中所示的路由器的实施方案的示意图。
图 3 是示出图 2 中所示的处理设备的实施方案的示意图。
图 4 是示出示例性分组的组成的示例图。
图 5A-C5 是示意图，每一个示出分组分类规则的实施方案。
图 6 是示出两级分组分类器（classifier）的实施方案的示意图。
图 7 是示出用于两级分组分类的方法的实施方案的示意图。
图 8 是示出示例性分类数据库的一部分的示意图。
图 9 是示出用于两级分组分类的方法的另一个实施方案的示意图。
图 10 是示出由多条规则组成的分类数据库的实施方案的示意图。
图 11A-11B 是示出将分类数据库组织成多个规则集的实施方案的示意图。
图 12A-12C 是示出两个部分重叠的过滤器的实施方案的示意图。
图 13A-13C 是示出两个完全重叠的过滤器的实施方案的示意图。
图 14 是示出在源地址和目的地址空间中两个并行查找的实施方案的示意图。
图 15 是示出分类数据库的过滤器中不存在的过滤器（non-existent filter）的图的示意图。
图 16A 是示出第一分类级数据结构的实施方案的示意图。
图 16B 是示出并行的最长前缀（prefix）匹配查找表的实施方案的示意图。
图 16C 是示出图 16A 的主查询表的实施方案的示意图。
图 17 是示出第二分类级数据结构的实施方案的示意图。
图 18 是示出用于两级分组分类的方法的实施方案的示意图，所述两级分组分类使用最具体的过滤器匹配和传输层共享。
图 19 是示出包括宽过滤器（wide filter）的过滤器数据库的实施方案的示意图。
图 20A 是示出第一分类级数据结构的另一个实施方案的示意图。
图 20B 是示出并行的最长前缀匹配查找表的另一个实施方案的示意图。
图 20C 是示出用于两级分组分类的方法的另一个实施方案的框图，所述两级分组分类使用最具体的过滤器匹配和传输层共享。
图 21 是示出用于创建和/或更新两级分类数据结构的方法的实施方案的框图。

具体实施方式

本文中公开了分组分类器的实施方案。下面在实现分组转发器（forwarder）（例如防火墙、基于 QoS 的转发器等等）的路由器环境中描述所公开的分组分类器的实施方案。但是，应该理解，所公开的实施方案在应用中不受此限制，此外，下面文字以及附图中所描述的分组分类器的实施方案可以应用于任何设备、系统和/或需要分组分类或其他通信分类的环境中。

图 1 中示出网络 100 的实施方案。网络 100 包括具有分组转发器 201 的路由器 200。路由器 200（和分组转发器 201）可以实现指定的安全策略、QoS 转发和/或资源预留，以及其他期望的基于策略的转发方案。为了区别属于不同流的分组和/或区别与不同流量类别相关联的分组，路由器 200 还包括分组分类器 600，所述分组分类器 600 包括被设计为实现期望的转发策略的规则组。下面更详细地描述分组分类器 600 的实施方案。路由器 200
（以及分组转发器201和分组分类器600）可以被实现在任何合适的计算系统或设备（或设备的组合）上，并且下面参照图2和相关文字描述了路由器200的一个实施方案。应该理解，路由器200可以包括其他组件（例如调度器（scheduler）、流量控制器等），为了明晰和易于理解，所述其他组件已经从图1中省略。

路由器200通过多条链路130（包括链路130a、130b…，130n）与多个节点（node）110和/或多个子网（subnet）120耦合。节点110包括任何可寻址设备。例如，节点110可以包括计算机系统或其他计算设备，例如服务器、桌面型计算机、膝上型计算机或手持式计算设备（例如个人数字助理或PDA）。子网120可以包括其他节点的集合，并且子网120还可以包括其他路由器和交换机。链路130a-n中的每一条都可以被建立在合适的介质上，例如无线、铜线、光纤或它们的组合，以支持通过任何合适的协议进行的信息交换。所述协议例如TCP/IP（传输控制协议/因特网协议）、HTTP（超文本传输协议）等等。

网络100可以包括任何类型的网络，例如局域网（LAN）、城域网（MAN）、广域网（WAN）、无线LAN（WLAN）或其他网络。路由器200还将网络100与其他网络（或其他多个网络）5耦合，所述其他网络例如因特网和/或其他LAN、MAN、LAN或WLAN。路由器200可以通过任何合适的介质使用任何合适的协议（例如TCP/IP、HTTP等）与所述其他网络5耦合，所述介质包括无线、铜线和/或光纤连接。

应该理解，图1中所示的网络100是要表示这样的系统的示例性实施方案，并且系统100可以具有任何合适的配置。例如，网络100可以包括额外的节点110、子网120和/或其他设备（例如交换机、路由器、网络中心（hub）等等），为了易于理解，所述额外的节点110、子网120和/或其他设备从图1中被省略。此外，应该理解，网络100可以不包括图1中所示的所有组件。

在一个实施方案中，路由器200包括任何合适的计算设备，在所述计算设备上可以（以硬件、软件、或硬件和软件的组合）实现分组分类器600。这样的计算系统的实施方案在图2中示出。

参照图2，路由器200包括总线205，各种组件耦合到所述总线205。总线205是要表示一种或多种将路由器200的组件互连的总线的集合，所述一种或多种总线例如系统总线、外设部件接口（PCI）总线、小型计算机系统接口（SCSI）总线等等。将这些总线作为单条总线205来表示是为了易于理解，并且应该理解路由器200不受此限制。本领域普通技术人员将意识到，路由器200可以包括任何合适的总线体系结构，并且可以包括数量的总线以及总线的组合。

与总线205耦合的是处理设备（或多个设备）300。处理设备300可以包括任何合适的处理设备或系统，包括微处理器、网络处理器、专用集成电路（ASIC）或现场可编程门阵列（FPGA），或类似设备。下面在图3和相关文字中示出了处理设备300的实施方案。
案。应该理解，尽管图 2 示例出单个处理器设备 300，但是路由器 200 可以包括两个或更多个处理设备。

路由器 200 还包括与总线 205 桥接的系统存储器 210，所述系统存储器 210 包括例如任何合适类型的和数量的随机访问存储器，例如静态随机访问存储器（SRAM）、动态随机访问存储器（DRAM）、同步 DRAM（SDRAM）或双倍数据率 DRAM（DDRDRAM）。在路由器 200 的操作期间，操作系统（或操作系统集）214、分组分类器 600 以及其他程序 218 可以驻留在系统存储器 210 中。图 2 的实施方案中，如下面所描述的，分组分类器的一部分——即分组分类器的第一级 600a（分组分类器）——包括可以驻留在系统存储器 210 中的软件例程（routine）集，而分组分类器的另一部分——即分组分类器的第二级 600b（分组分类器）——以硬件实现。但是，应该理解，图 2 的实施方案仅仅表示所公开的分组分类器的一个实现，并且所公开的分组分类器可以以任何合适的软件和/或硬件配置实现。例如，在另一个实施方案中，第二分类级 600b 可以包括驻留在系统存储器 210（并且可能存储在存储设备 230 中）中的软件例程集。

路由器 200 还可以包括与总线 205 桥接的只读存储器（ROM）220。在操作期间，ROM 220 可以存储用于处理设备的临时指令和变量。路由器 200 还可以包括与总线 205 桥接的存储设备（或多个存储设备）230。存储设备 230 包括任何合适的非易失性存储器，例如硬盘驱动器。分组分类器 600（例如分组分类器 600 的第一级 600a），以及操作系统 214 和其他程序 218（例如分组转发器 201 的软件实现）可以被存储在存储设备 230 中。此外，用于访问可移除存储介质（例如软盘驱动器或 CD ROM 驱动器）的设备 240 可以与总线 205 桥接。

路由器 200 还可以包括与总线 205 桥接的一个或多个输入设备 250。普通输入设备 250 包括键盘、诸如鼠标的定点设备，以及其他数据录入设备。一个或多个输出设备 260 也可以与总线 205 桥接。通用输出设备包括视频显示器、打印设备、音频输出设备以及状态指示器（例如 LED）。

路由器 200 还包括与总线 205 桥接的网络/链路接口 270。网络/链路接口 270 包括任何合适的能够将路由器 200 与所述其他网络（或多个网络）5 桥接并且能够将路由器 200 与链路 130a-n 中的每一条桥接的硬件、软件，或软件和硬件的组合。

应该理解，图 2 中示出的路由器 200 是要表示这样的设备的示例性实施方案，并且该系统可以包括很多额外的组件。为了清晰和易于理解，所述额外的组件被省略。作为实施方案，路由器 200 可以包括 DMA（直接存储器访问）控制器、与所述处理设备 300 相关联的芯片组、额外的存储器（例如缓存存储器），以及额外的信号线和总线。此外，应该理解，路由器 200 可以不包括图 2 中所示的所有组件。

在一个实施方案中，分组分类器 600 或分组分类器的一部分包括在计算设备上运行的
指令集（即软件应用），所述计算设备例如图 2 的路由器 200 或其他合适的计算设备。指令集可以以本地的方式被存储在存储设备 230（或其他合适的程序存储器）中。可替换地，所述指令可以被存储在远程存储设备（图中未示出），并且通过网络 100（或通过另一网络 5）来访问。在操作期间，指令集可以在处理设备 300 上被执行，其中所述指令（或其一部分）可以驻留在系统存储器 210 中。

在另一个实施方案中，分组分类器 600（或分组分类器的一部分）包括存储在机器可访问介质上的指令集，所述机器可访问介质例如磁介质（例如软盘或磁带）、光可访问盘（例如 CD-ROM 盘）、闪存存储器设备等。为了在例如图 2 的路由器 200 上运行分组分类器 600，用于访问可移除存储介质的设备 240 可以访问所述机器可访问介质上的指令，并且随后所述指令可以在处理设备 300 中被执行。在该实施方案中，指令（或其一部分）可以被再次下载到系统存储器 210 中。

在再一个实施方案中，分组分类器 600 或分组分类器的一部分以硬件实现。例如，分组分类器 600（或其一部分）可以用内容可寻址存储器（CAM）来实现。在另一个实施方案中，分组分类器 600 可以用硬件和软件的组合来实现。

在一个下面将详细描述的特定实施方案中，分组分类器 600 包括两级分组分类系统，并且该两级分类方案既可以以硬件实现又可以以软件实现。两级分组分类器包括第一级 600a 和第二级 600b（见图 2）。在一个实施方案中，第一级 600a 以软件实现，并且第二级 600b 以硬件实现。

如之前注意到的，在图 3 和相关文字中示出处理器设备 300 的实施方案。但是，应该理解，图 3 中所示处理设备 300 仅仅是处理设备的一个实施方案，所公开的分组分类器 600 的实施方案可以被实现在所述处理设备上。本领域普通技术人员将意识到，所公开的分组分类器 600 的实施方案可以被实现在很多其他类型的处理系统和/或处理器体系结构上。

现在参照图 3，处理设备 300 包括本地总线 305，各种功能单卷耦合到所述本地总线 305。总线 305 是要表示一种或更多种将处理设备 300 的各种功能单元互连的芯片上（on-chip）总线的集合。将这些本地总线 305 表示为单条总线 305 是为了易于理解，并且应该理解，处理设备 300 不受此限制。本领域普通技术人员应该意识到，处理设备 300 可以包括任何合适的总线系统结构，并且可以包括任意数量的总线和总线的组合。

核心（core）310 和多个处理引擎（engine）320（例如处理引擎 320a、320b...320k）与本地总线 305 耦合。在一个实施方案中，核心 310 包括可以执行操作系统 214 的通用处理系统。核心 310 还可以控制处理设备 300 的操作，并且可以执行各种管理功能，例如向处理引擎 320 分发用于执行的指令。所述处理引擎 320a-k 中的每一个都可以包括任何合适的处理器系统，并且每一个都可以包括算术逻辑单元（ALU）、控制器以及多个寄存器（用于在读/写操作中存储数据）。此外，在一个实施方案中，每个处理引擎 320a-k 提供多个
执行线程（例如四个）。

还与本地总线 305 耦合的是芯片上存储器子系统 330。尽管芯片上存储器子系统 300 被示为单个单元，但是应该理解它可以——并且在实践中可能的确——包括多个不同的存储器单元和/或存储器类型。例如，这样的芯片上存储器可以包括 SDRAM、SRAM 和/或闪存存储器（例如 FlashROM）。应该理解，除了芯片上存储器以外，处理设备 300 可以与芯片外（off-chip）存储器（例如 ROM 220、芯片外缓存（cache）存储器等等）耦合。

处理设备 300 还包括与本地总线 305 耦合的总线接口 340。总线接口 340 提供与路由器 200 的其他组件（包括总线 205）的接口。为了简明，总线接口 340 被示为单个功能单元；但是应该理解，在实践中，处理设备 300 可以包括多个总线接口。例如处理设备 300 可以包括 PCI 总线接口、IX（因特网交换）总线接口等等，并且总线接口 340 是要表示一种或更多种这样的接口的集合。

应该理解，参照图 3 所示出和描述的处理设备 300 的实施方案仅仅是可能发现与所公开的分组分类器的实施方案一起使用的处理设备的一个实施例，并且处理设备 300 可以包括图 3 所示组件之外的其他组件，为了明晰和易于理解，所述其他组件被省略。例如，处理设备 300 可以包括其他功能单元（例如指令解码单元、地址翻译单元等）、热管理系统、时钟电路、额外的存储器以及寄存器。此外，应该理解处理设备可以不包括图 3 中所示的所有部件（element）。

现在参照图 4，所示为可以在路由器 200 处（从其他网络 5）接收的分组 400 的实施例。分组 400 包含头部 410 和有效载荷（payload）（或数据）450。头部 410 可以具有任何合适的格式，并且图 4 中示出的分组 400 示出与 TCP/IP 协议相关的头部的实施例。例如，参见因特网工程任务组请求评审（IETF RFC）791，Internet Protocol（因特网协议）（1981），以及 IETF RFC 793，Transmission Control Protocol（传输控制协议）（1981）。在图 4 中，头部 140 包括多个字段，所述多个字段包括字段 420a、420b…420x。通常，字段 420a-x 包括关于分组 400 的标识信息。以实施例的方式，头部 410 可以包括协议 420i（例如 TCP）、源地址 420k、目的地址 420j、源端口 420m 以及目的端口 420n。每一个源地址和目的地址 420k、420i 可以包括三十二（32）个位（bit），每个源和目的端口 420m、420n 可以包括十六（16）个位，每个协议 420i 可以包括八（8）个位。本领域普通技术人员应该意识到，这些仅仅是可以被包括在分组的头部中的信息类型的一些实施例，并且分组头部 410 可以按照手边具体的硬件和/或应用的要求包括任何信息。此外，应该理解，分组的格式不限于结合图 4 所示出和所描述的格式（例如头部字段的宽度可以变化、字段的类型和数量可以变化等等）。

在本文中通信被一般性地称作“分组”。但是，应该理解，所公开的实施方案可应用于任何类型的通信（例如分组、信元（cell）、帧（frame）等等），而与格式和内容无关。
参照图 5A，示出分组分类规则 500 的实施方案。一般地，规则 500 指明支持特定流的准则组，满足所述准则的分组属于所述特定的流。规则 500 包括多个字段，包括字段 510a（FIELD 1）、510b（FIELD 2）、...510n（FIELD N）。规则可以包括任何合适数量的字段 510a-n，且在文本中规则中字段的数量被称作维数（即规则 500 具有的维数为 N）。在一个实施方案中，每个字段 510a-n 对应于分组的头部中的字段，例如源地址或目的地址。但是，在其他实施方案中，规则的组成部分（component）510a-n 可以包括其他信息，例如应用头部字段、链路标识信息、日期时间等等。一般地，如果对于每个字段 510a-n 来说，分组头部中的对应字段与规则的字段匹配，则该分组与规则 500 “匹配”（分类规则的字段通常被表示为值的范围，并且如果某个值被包括在对应此规则中某个字段的范围内，则该值与该字段匹配）。规则 500 还包括要被施加到每个匹配该规则的分组上的相关动作 520（例如接受、阻挡等）。规则 500 还与优先级 530 相关联。

参照图 5B，示出分组分类规则的另一个实施方案 501。规则 510 包括源地址字段 510a 和目的地址字段 510b，以及多个传输层字段 510c。作为实施例，传输层字段 510c 可以包括协议 515a、源端口 515b 和目的地端口 515c。传输层字段 510c 不限于前面提到的字段，并且传输层字段 510c 可以包括其他字段 515d-k（或者它可以包括更少的字段）。其他可能的传输层字段包括例如 TCP SYN 标志（flag）和 RSVP 头部字段（参见例如 IETF RFC 2205，Resource Reservation Protocol（RSVP）-Version 1 Functional Specification（资源预留协议（RSVP）-第 1 版功能规范）（1997））的协议字段等等。规则 510 还包括相关动作 520 和优先级 530。

图 5C 中所示为图 5B 中示出的规则的实施方案。图 5C 的规则 502 指明源 IP（因特网协议）地址 510a 等于 “128.128.*”，目的 IP 地址 510b 等于 “128.67.120.84”，协议 510c 等于 “TCP”，源端口 510d 等于 “80”，目的地端口 510e 等于 “*”，其中字符 “*” 代表“通配符（wild card）”（即任何值都能够在通配符匹配）。动作 540 是“阻挡（block）”（即任何满足该规则的分组都不被允许，并且规则 502 具有等于 “2” 的优先级 540。当然，应该理解，图 5C 仅仅呈现出规则的一个实施方案，并且规则可以包括任何合适的头部字段数量和类型（即规则可以是任意维的）。

规则的每个字段都可以被表示为精确匹配（例如源端口等于 “80”），前缀（例如源地址为 “128.128.*”）或范围说明（例如源端口 “≤1023”）。但是，一些范围（例如 “>1023” 的源端口）不能用前缀来表示，并且这样的表达式可以被分解为一组前缀。例如范围 “>1023” 可以用下面一系列的前缀（二进制格式）来描画：
“000001**********”；“0001**********”；“001**********”；“01**********”；以及 “1**********”。因此，具有字段 “>1023”的规则可以被扩展为六个不同的规则，每个规则对应这六个不同的前缀中的每一个，所述六个不同的前缀组成范围说明 “>1023”。在此应该注意，一般来说，K 位的范围可以被分解为最多（2K-2）个前缀。
图6中所示为分组分类器600的实施方案。分组分类器600将分类过程分裂成两个级。分组分类器600包括第一级600a（STAGE1）和第二级600b（STAGE2）。在第一级600a中，分组基于其网络路径被分类，并且在第二级600b中，分组基于一个或更多个其他字段（例如传输层字段）被分类。与第一级600a相关联的是逻辑610和数据结构1600，与第二级600b相关联的是逻辑620和数据结构1700。

第一级逻辑610包括任何合适的能够基于分组的网络路径对分组进行分类的软件、硬件或软件和硬件的组合。在一个实施方案中，第一级逻辑610基于接收的分组的源地址和目的地址确定结果，并且该结果被提供给分类器的第二级600b。下面描述了基于分组的网络路径对分组进行分类的各种实施方案。网络路径通常用源地址和目的地址（例如源IP地址和目的IP地址）来表示。但是，应该理解，网络路径的表示并不限于源-目的地址对，并且可以使用其他可替换的准则来标识网络路径，例如可以使用多协议标签交换（MPLS）标签（参见例如IETFRFC3031，MultiprotocolLabelSwitchingArchitecture（多协议标签交换）（2001））、源IP地址和目的多播组的组合等等来标识。第一级数据结构1600（在下面也将详细描述）可以被存储在任何合适的存储器中，包括SRAM、DRAM、SDRAM、DDRAM以及其他存储器类型。

第二级逻辑620包括任何合适的能够基于被包括在接收的分组的头部中的传输层字段（或其他字段）对接收的分组进行分类的软件、硬件或软件和硬件的组合。在一个实施方案中，第二级逻辑620从第一级接收分类的结果，并且基于该结果和接收的分组的其他字段（例如传输层字段）确定要应用于所述分组或要以其他方式执行的动作。下面描述了基于被包括在分组头部中的一个或更多个传输层字段（或其他字段）对分组进行分类的方法的各种实施方案。第二级数据结构1700可以被存储在任何合适类型的存储器中，例如CAM、SRAM、DRAM、SDRAM，或其他类型的存储器。第二级数据结构1700也在下面更详细地被描述。

如上面所提到的，在一个特定的实施方案中，分组分类器600以软件和硬件的组合实现。更具体地，第一分类级600a以软件实现，而第二分类级600b以硬件实现。在这个实施方案中，第一分类级600a可以包括指令集，所述指令集被存储在存储器（例如图2中示出的系统存储器210和/或图3中示出的芯片上存储器子系统330）中，并且在处理设备（例如图3的处理设备300）上被执行，而第二级600b可以用CAM（或其他硬件配置）来实现。

在图7中进一步示出前面提到的可以被实现在图6的两级分组分类器600上的两级分类方案，图7示出用于两级分组分类的方法700的实施方案。参照图7中的框710，接收分组，并且如框720所阐述的，基于所述分组的网络路径来对所述分组进行分类。一般地，分组的网络路径被表示为包括在分组的头部中的源地址和目的地址。然后，基于被包括在接收的分组的头部中的一个或更多个其他字段对分组进行分类，这在框730中阐述。基于两级分类的结果（参见框720、730）识别出最高优先级动作，并且将该动作应用于所
述分组，如框 740 中所阐述。

在因特网以及很多其他大的网络中，通常在整个网络中存在很多可能的路由，但是存在着相对少的应用。因此，遵循这样的规则，不同的网络路径的数量一般远远大于应用的数量。这些观察是通过实际分析数据库的研究得出的，这表明在分类规则集中找到的源-目的地址对数量一般远远大于其他地址的数值（例如因特端口数量和协议的传输层地址）。这些研究还表明很多不同的源-目的地址对使用相同的传输层字段（或其他）集，并且与所述集中每个成员（member）相关的相对优先级和动作在所述的集的每次发生中都是相同的。此外，每个集中的项数一般较小。

图 8 中示出分类数据库中的源-地址对常规地使用相同的传输层字段集的事实。参照图 8 显示示例性分类数据库 800。分类数据库 800 包括多个分类规则，每个规则包括源 IP 地址 810a、目的 IP 地址 810b、协议 810c、源端口 810d 和目的端口 810e，以及相关联的动作 820 和绝对优先级 830a。图 8 中示出两个规则组 801、802，并且第一组的每条规则具有相同的网络路径，如源和目的 IP 地址 810a-b 所表示的。第二组的每条规则具有相同的网络路径（但是与第一组的网络路径不同）。在图 8 中还示出规则组（801 或 802）内的每条规则的相对优先级 830b。尽管两个规则组 801、802 各自具有不同的源-目的地址对，但是这些规则组共享相同的传输层字段集、动作和相对优先级（即规则组 801 的协议、端口说明、动作和属性对于组 802 重复）。在本文中，（一个或更多个）传输层字段集、动作和相对优先级的组合被称为“三元组（triplet）”。

参照图 6，在分组分类的第一级 600a 中，N 维分类问题被降低到两维问题，因为第一级中的分类可以基于源地址和目的地址被执行，这大大地简化了分类的该级。尽管第二级 600b 可以包括基于任意数量字段（例如三个或更多个）的分类，但是通过利用前面提到的现实世界的分类数据库的特性可以相当地降低分类问题的复杂度。具体来说，通过利用很多源-目的地址对共享相同的三元组成（即传输层字段集、动作和相对优先级），需要检查的项数的数量以及存储器访问次数可以被实质地减少。在本文中，将多个源-目的地址对与一组不同的传输层（或其他）字段集——即因为这些源-目的地址对中的每一个使用相同的三元组——相关的概念被称为“传输层共享”（“TLS”）。尽管传输层字段组潜在地可以被多个源-目的地址对共享，但是每个独特的传输层字段组可以被存储一次，由此降低分组分类系统的存储需求。

如上面注意到的，通过将多维分类问题降低到两维分类问题，简化了两级分类方案的第一级 600a。但是，有可能——并且在实践中极有可能——分组与分类数据库中的多条规则匹配，并且因此第一级 600a 将返回多条匹配。多条匹配可能发生，至少部分是由于分类数据库中所有规则的源-目的对的重叠性质，其次是由于源-目的对可以与任意优先级相关联。寻找所有可能的匹配规则可能大大地增加第一分类组中所需的存储器访问次数。此外，当从第一分类组返回多条匹配时，合并分类的第一级 600a 和第二级 600b 之间的结果变得困难。因此，在另一个实施方案中，为了简化分组分类的第一级 600a 并增加其效率，
从第一级返回单个、最具体的匹配。在一个实施方案中，“最具体的匹配”是指这样的匹配，它提供最大量的关于分组的网络路径的信息（例如，对于 IP 网络，最具体的匹配是覆盖分组的所有过滤器的交集）。在本文中，确定并从第一分类级 600a 返回单个匹配过滤器的作被称为“最具体的过滤器匹配”（“most specific filter”或“MSFM”）。

参照图 9，示出用于两级分组分类的方法 900 的另一个实施方案。用于两级分组分类的 方法 900 实现传输层共享和最具体的过滤器匹配。参照图 9 的框 910，接收到分组。如框 920 所阐述的，在分组的第一级，基于所述分组的源和目的地址来对该分组进行分类，以寻找单个、最具体的匹配。在框 930 所阐述的分类的第二级中，基于一个或更多个使用传输层共享的传输层字段来对分组进行分类。如框 940 所阐述的，基于分类的两级的结果，确定最高优先级的运作，并将此动作应用于该分组。应该理解，在另一个实施方案中，在第一级使用最具体的过滤器匹配而在第二级不使用传输层共享，然而在进一步的实施方案中，在第二级使用传输层共享而在第一级不使用最具体的过滤器匹配。

现在将更详细地描述最具体的过滤器匹配（MSFM）和传输层共享（TLS）。描述将 以对 TLS 的描述开始，随后描述 MSFM。

典型的分类数据库包括规则列表。这在图 10 中示出，图 10 示出包括多条规则 1005
的分类数据库 1000，所述多条规则 1005 包括规则 1005a、1005b，…，1005y。规则 1005a-y 中的每一条包括例如源地址 1010a、目的地址 1010b 和一个或更多个传输层字段 1010c（或其他字段），以及相关联的动作 1020 和优先级 1030。如之前注意到的，对于每条规则 1005，传输层字段 1010c、动作 1020 和优先级 1030 的组合可以被称为“三元组”1040。如上面讨论的，典型的分类数据库包括比之源-目的对来说相对少的独特的传输层字段集，并且多个源-目的对通常共享典型的分类数据库中的传输层字段组组成的相同的组（参见图 8 和相关文字）。为了利用分类数据库的该特性，数据库 1000 的规则 1005 被分组为规则集。规则集包括那些具有相同的源和目的地址对的数据库的规则。例如，参照图 8，组 801 中的那些规则可以包括第一规则集（即它们共享源地址“128.128.10.5”和目的地址“172.128.*”），并且组 802 中的那些规则可以包括第二规则集（即它们共享源地址“128.128.10.7”和目的地址“172.128.*”）。

在图 11A 中示出将分类数据库分割成多个规则集的实施例。参照图 10，分类数据库 1100 已被组织成多个规则集 1150，包括规则集 1150a、1150b，…，1150q。每个规则集 1150a-q 包括一条或多条规则，其中给定规则集 1150 中的每条规则共享与该规则集中的所有其他规则相同的源和目的地址对。如图 11A 中所示，一个实施方案中，数据库 1100 的规则被排序，从而每个规则集 1150a-q 的规则占用数据库中连续的空间。同样，规则集 1150 内的规则可以占用连贯的（consecutive）优先级级别；然而，在另一个实施方案中，规则集可以由不占用连贯的优先级级别的规则组成。可以认为规则集 1150a-q 各自与过滤器 1160 相关联（即规则集 1150a 与过滤器 1160a 相关联，规则集 1150b 与过滤器 1160b 相关联，等等），其中用于规则集 1150 的过滤器 1160 包括用于该规则集的源-目的对。
应该理解，图11A的实施方案仅仅是分类数据库的规则可以组成规则集的一种示例性方式。

图11A中的每个规则集1150（以及每个对应的过滤器1160）将包括相关联的三元组组成的组，每个三元组包含多个传输层（或其他）字段、动作和相对优先级。再次回到图8，每个规则集1001、1002包含三个与它相关联的三元组组成的组，并且在图8的实施例中，正巧每个规则集1001、1002包含相同的三元组组成的组（即规则集1001、1002的目的地对分别共享相同的三元组）。在本文中，与任何规则集相关联的三元组和过滤器组成组，所述组被称为“容器（bin）”，并且在这种情况下是“小容器”（区别于下面描述的“大容器”）。因此，如图11A所示，每个规则集1150和对应的过滤器1160与三元组的小容器1170相关联（即规则集1150a和过滤器1160a与小容器1170a相关联，等等）。在被包括在规则集中的多条规则间共享由各种传输层字段集组成的组（即小容器）导致传输层共享的概念。

与图11A的数据库1100相关联的小容器1170在图11B中进一步示出。参照图11B，数据库1100包含一批与之相关联的小容器1170。每个小容器1170包含一个或更多项1172，每个项1172包含一个或更多个传输层（或其他）字段集1174、相对优先级1175和动作1176（即三元组）。如图11B所示，小容器1170可以进一步从逻辑上组织成“大容器”1180，每个大容器1180包含一批两个或更多个小容器1170。下面将描述大容器1180的创建。

下面将描述用于基于传输层字段分类分组的第二级数据结构1170。现在我们将注意力转向对最具体的过滤器匹配的讨论。

如之前提示的，当基于分组的网络路径分类分组时，极有可能分组会与分类数据库中的多条规则匹配。为了增加第一分类级600a的效率——例如为了降低所需要的存储器访问次数——并且为了简化将第一级的结果与第二分类级600b合并的操作，需要从第一级返回单个、最具体的匹配。

应该注意到在我们的讨论当中，在这一点上使用术语“过滤器”而不使用“规则”，因为使用在这里，术语“过滤器”是指与规则集相关联的目的地址对（参见图11A-11B和上面的相关文字）。但是，应该理解，术语“过滤器”和“规则”通常是可互换使用的。

基于分组的源和目的地址被分类的分组可以匹配多个过滤器，至少部分是由于分类数据库中的过滤器的重叠性质。这在图12A到12C以及图13A到13C中示意性地示出。首先参照图12A，可以认为规则的源和目的地址是二维空间中的一个区域（即矩形、线或点中的任意一种）。在图12A中，分类数据库的第一过滤器1210占用被标记为F1的矩形所示的二维空间。过滤器F1与三元组组成的小容器1215（B1）相关联。参照图12B，数据库的第二过滤器1220（F2）也占用源和目的地址区域中的一个区域，并且该过滤器F2
与由传输层字段组成的小容器 1225 (B2)。

在图 12C 中示出过滤器 F1 和 F2 两者，并且可以观察到由这两个过滤器重叠或交叉所定义的区域，F1 和 F2 的交叉由符号 1290 指示。图 12A-12C 中示出的过滤器 F1 和 F2 称为“部分重叠”。如果分组 (P) 1205 具有的源-目的地址对落入过滤器 F1 和 F2 的交叉 1290 中，则该分组与两个过滤器匹配（即过滤器 F1 和 F2 至少包括由分组 P 定义的共同点）。同样的，分组 P 以及其他落入过滤器 F1 和 F2 的交叉中的分组与小容器 B1 和 B2 的并集相关联，所述小容器 B1 和 B2 分别与 F1 和 F2 相关联。在本文中，与两个或更多个过滤器的交叉相关联的小容器的并集被称为“大容器”。在图 12C 中，包括容器 B1 和 B2 的并集的大容器由符号 1295 指示。

过滤器还可以“完全重叠”，该场景在图 13A 到 13C 中示出。第一过滤器 1310 (F1) 占用图 13A 中示出的两维空间中的区域，第二过滤器 1320 (F2) 占用图 13B 中示出的空间中的区域。过滤器 F1 包括相关联的小容器 1315 (B1)，并且过滤器 F2 包括相关联的小容器 1325 (B2)。接下来参照图 13C，示出了过滤器 F1 和 F2，并且这些过滤器在交叉处重叠（在该实例中交叉 1390 等于 F2 定义的区域）。因为 F2 定义的区域（以及交叉 1390）被完全包含在过滤器 F1 的两维空间中，所以过滤器 F1 和 F2 被称为“完全重叠”。“落入交叉 1390 的分组 (P) 1350 与过滤器 F1 和 F2 两者匹配，并且该分组 P（以及任何落入该交叉 1390 的分组）与大容器 1395 相关联，所述大容器 1395 由小容器 B1 和 B2 的并集组成。从图 12A-12C 以及 13A-13C 中可以观察到，具有位于两个（或更多个）过滤器的交叉处源地址和目的地址的分组与两个过滤器匹配。为了达到从分类的第一级仅返回单个匹配的目标，所有过滤器的交叉都被包括在过滤器数据库（即第一级数据结构 1600）中。此外，每个过滤器交叉都与构成该交叉的各个过滤器的传输层字段的并集相关联。换言之，如上所述，每个过滤器交叉都与包括两个或更多个小容器的并集的容器相关联。理论上，将所有过滤器交叉添加到查找表中可能需要大的存储容量。但是，对实际分类数据库的研究表明，在实践中，存储过滤器交叉所需的额外容量远远小于理论上。

为了基于分组的源和目的地址分类分组，问题变成查找分组所在的最小过滤器交叉的问题（或者如果分组不落在交叉，则简化为分组所在的过滤器）。为了找到这个最小的过滤器交叉，分类的第一级（两分类问题）被分裂成两个一维的查找。这在图 14 中示意图地示出，图 14 中示出第一维 1410 和第二维 1420。在第一维 1410 中，将接收的分组的源地址与源地址查找数据结构 1412 的项进行比较，并且如果找到匹配，则返回索引 1414 (I1)。类似地，在第二维 1420 中，将分组的目的地址与目的地址查找数据结构 1422 的项进行比较，并且如果找到匹配，则返回第二索引 1424 (I2)。然后，两个索引 1414, 1424 (I1 和 I2) 被组合以形成关键字 1430，所述关键字 1430 可以被用来查询另一个查找数据结构（例如哈希表）。在一个实施方案中，在第一维和第二维 1410, 1420 的查找数据结构 1412、1422 上分别执行的查找是通过使用最长前缀匹配 (LPM) 来完成的。在再一个实施方案
中，两个维 1410、1420 中的查找是并行执行的。人们相信，在源和目的上使用并行和独立的查找会导致对少的存储器访问次数，并且会提出合理的存储要求。

图 14 中示出的并行 LPM 查找方案将返回分组所在的最小的过滤器交叉，或是该方案将返回“不存在的”过滤器。不存在的过滤器包括一个过滤器的源地址和不同的过滤器的目的地址。不存在的过滤器是由于独立地执行源和目的地址的查找而导致的（尽管对于一个实施方案来说，这些查找是并行执行的）。

参照实施例可以很好地理解不存在的过滤器的概念。参照图 15，数据库包括由字母 A、B、C、D 和 E 指示的五个过滤器，并且这些过滤器在两维地址空间 1500 中示出。过滤器 A 覆盖整个源和目的地址空间，并且该过滤器可以用“*，*”来指示（即它在源和目的地址中都包括通配符“*”）。过滤器 B 的形式为“*，Y*”，并且这种形式（即“*，Y*”或“X*，*”）的过滤器被称为“部分指明的过滤器（partially specified filter）”。过滤器 C、D 和 E 被称为“完全指明的过滤器（fully specified filter）”，并且这些过滤器的形
式为“X*，Y*”。以实施例的形式，过滤器“SRC ADD 128.172.*/DST ADD *”是部分指明的过滤器，而过滤器“SRC ADD 128.172.*/DST ADD 128.128.*”是完全指明的过滤器。过滤器 A、B、C、D 和 E 的源和目的地址的组合形成二维空间中的十二个不同的区域。第一组五个区域与过滤器 A、B、C、D 和 E 对应。由 R1 到 R7 指示的其他七个区域（用虚线示出）与不存在的过滤器对应。区域 R1 到 R7 是由一个过滤器的目的地址和另一个过滤器的源地址形成的，并且如上面注意到的，这些区域是由于独立地执行源和目的地址的查找而创建的。例如，区域 R4 是通过组合过滤器 E 的源地址和过滤器 D 的目的地址创建的，这就产生了不存在的过滤器。图 15 中示出的图被通称为交叉组合（cross-producting）表。

为了确保图 14 的并行查找方案返回结果，看起来所有不存在的过滤器 R1 到 R7 都需要被录入到过滤器数据库中。多项现有技术（例如交叉组合表方案）建议将所有不存在的过滤器添加到过滤器数据库中。但是，对实际分类数据库的观察表明，不存在的过滤器的数量可能非常大，并且因此需要最小化被添加到过滤器数据库中的不存在的过滤器的数量。因此，在一个实施方案中，仅仅在分类数据结构中包括所有可能的不存在的过滤器的子集。下面更详细地描述了避免将所有可能的不存在的过滤器添加到分类数据结构中（以及不存在的过滤器的子集代替所有可能的不存在的过滤器被放置到数据结构中）的方式。

参照图 15，可以观察到区域 R1 到 R7 中的很多可以被聚合成少量的其他过滤器。具体来说，可以完全覆盖区域 R1、R3 和 R4 的最小的过滤器是过滤器 A，也即整个两维空间。对于任何分组的源或目的地址落入区域中的一个的搜索将返回被包括在区域 A 中的过滤器的源或目的地址。因此，不存在的过滤器 R1、R3 和 R4 可以被聚合成过滤器 A 中，并且可以过滤器数据库中移除针对 R1、R3 和 R4 的各个项，假设存在针对过滤器“*，*”的单独的项。类似地，过滤器 B 完全覆盖区域 R5、R6 和 R7，并且这些不存在的过滤器可以与针对过滤器 B 的项聚合。对于任何位于区域 R5、R6 和 R7 中的一个的分组来说，
对于该分组的目的地址的搜索将返回过滤器B的目的地址。因此，不需要针对R5、R6和R7的单独的数据项，因为在分类数据库中提供了针对部分指明的过滤器B的单独的项。

但是，区域R2不能被合并到任何其他过滤器中。由过滤器D的源地址和过滤器E的目的地址形成的区域R2被完全指明的过滤器——即过滤器C——完全覆盖。不存在的过滤器R2与其他不存在的过滤器的区别在于，它是唯一一个被完全包括在完全指明的过滤器中的区域。不存在的过滤器R2不能与过滤C或任何其他项聚合并且针对该过滤器的项必须被放置到过滤器数据库中。在本文中，诸如R2的不存在的过滤器也被称为“指示器过滤器”。指示器过滤器与这样的传输层字段相关联，所述传输层字段对应于完全覆盖该指示器过滤器的最小可能过滤器交叉。以实施例的方式，对于图15中示出的过滤器集来说，完全覆盖区域R2的过滤器是过滤器A和C，并且这两个过滤器的交叉简单地为过滤器C。因此，指示器过滤器R2所关联的传输层字段集与过滤器所关联的传输层字段集相同。

一些额外的观察帮助了第一级数据结构1600的发展。一般来说，在分类数据库的过滤器之间存在三种部分重叠的来源，包括：（1）在部分指明的过滤器（即形式为“X*, *”或“*, Y*”的过滤器）之间创建的部分重叠；（2）在完全指明的过滤器（即形式为“X*, Y*”的过滤器）之间创建的部分重叠；以及（3）在部分指明的过滤器和完全指明的过滤器之间创建的部分重叠。注意到，每个源维（source dimension）中具有通配符的部分指明的过滤器与所有在目的维中具有通配符的部分指明的过滤器创建部分重叠，并且由这样的部分指明的过滤器的交叉创建的部分重叠的数量等于分别在每个源和目的维中的部分指明的过滤器的数量的积。理论上，由部分指明的过滤器的交叉引起的部分重叠的数量可能很大。另一方面，完全指明的过滤器相互之间创建数量不多的部分重叠，这是由于在实践中大多数完全指明的过滤器是二维地址空间中的直线段或点。

如之前的段落所指出的，部分指明的过滤器通常是典型分类数据库中的过滤器中部分重叠的主要来源。但是，部分指明的过滤器常常代表分类数据库中所有过滤器的总数量的小部分，因为网络管理员常常指明应用于特定地址域间交换的流量的规则。因此，在实践中，由部分指明的过滤器引起的部分过滤器重叠的数量比理论上最坏的情况显著地小。此外，如上面所注意到的，完全指明的过滤器在过滤器间创建少量部分重叠。因此，在实际分类数据库中出现的部分重叠的数量一般比理论上预期发生的数量少得多。

在这一点上，应该注意到我们没有关注在图13A-13C中示出的完全重叠过滤器。如上面所阐述的，在一个实施方案中，最长前缀匹配（LPM）被用于在MSFM中执行的两个一维搜索中的每一个。如果过滤器完全重叠，这些过滤器的交叉等于这些过滤器中的一个，并且LPM搜索将识别此过滤器。因此，在一个实施方案中，第一级数据结构1600不需要考虑完全重叠过滤器。

上面的观察和讨论（例如见图6-15和相关文字）介绍了可以用于构建第一级数据结构
构 1600 以及第二级数据结构 1700 的“建构模块（building block）”，现在描述这两种数据结构的实施方案。下面还要介绍搜索第一和第二级数据结构 1600、1700 以对分组进行分类（即识别要应用于所述分组的动作）的方法的实施方案。

现在参见图 16A，示出第一级数据结构 1600 的实施方案。第一级数据结构 1600 包括并行 LPM 数据结构 1601 和转发表 1602。如上面所注意到的，MSFM 方案采用分别在源和目的地址上执行的两个一维查找，如上面关于图 14 所示出和描述的。因此，并行 LPM 数据结构包括源地址查找数据结构 1610 和目的地址查找数据结构 1620。在一个实施方案中，源和目的地址查找数据结构 1610、1620 被实现为特里数据结构（trie data structure）。但是，本领域普通技术人员将意识到可以使用其他替换的数据结构来实现源和目的地址查找数据结构 1610、1620。

在图 16B 中进一步示意图地示出并行 LPM 数据结构 1601 的实施方案。源地址查找数据结构 1610 包括多个项 1612，每个项 1612 指明源前缀 1614a、过滤器类型 1614b 以及索引值 1614c（或其他标识符）。类似地，目的地址查找数据结构 1620 包括多个项 1622，每个项 1622 指明目的前缀 1624a、过滤器类型 1624b 以及索引值 1624c（或其他标识符）。对于查找数据结构 1610、1620 来说，过滤器类型 1614b、1624b 指示项 1612、1622 是与完全指明的过滤器相关联还是与部分指明的过滤器相关联。当在并行 LPM 数据结构 1601 上执行搜索时，返回四个索引（或其他标识符），所述四个索引包括源地址查找数据结构 1610 中的与匹配完全指明的过滤器相关联的第一索引 1691（I1）、目的地址查找数据结构 1620 中的与匹配完全指明的过滤器相关联的第二索引 1692（I2）、源地址查找数据结构中的与匹配部分指明的过滤器相关联的第三索引 1693（I3）以及目的地址查找数据结构中的与匹配部分指明的过滤器相关联的第四索引 1694（I4）。如上面所注意到的，在一个实施方案中，在源和目的地址维中的查找以并行的方式执行。

与完全指明的过滤器相关联的头两个索引 1691 和 1692（I1 和 I2）被组合以创建关键字 1690。如下面将会描述的，与完全指明的过滤器相关联的关键字 1690 以及与部分指明的过滤器相关联的第三和第四索引 1693、1694 被用来搜索转发表 1602。在此交界点（juncture）区分完全指明和部分指明的过滤器的理由是，匹配过滤器应该是部分指明的过滤器并且最长前缀匹配应该被用在分类的第一级中，正在找寻的部分指明的过滤器可能不被识别（即，识别的匹配源或目的前缀可以比实际匹配过滤器对应的前缀 “长”）。

在一个实施方案中，如图 16A 中所示出，转发表 1620 包括主表 1630 和两个次表 1640a、1640b。主表 1630 包括针对完全指明的过滤器、完全指明的过滤器的交叉以及指示器过滤器的项。再一次，指示器过滤器是由不同源-目的对的源和目的前缀形的区域，并且所述区域不能和完全指明的过滤器或过滤器交叉聚合（例如参见图 15，上面讨论的区域 R2）。次表 1640a-b 中的一个包括针对在源维中具有通配符的部分指明的过滤器的项，而次表 1640a-b 中的另一个包括针对在目的维中具有通配符的部分指明的过滤器的项。次表 1640a、1640b 不包括指示器过滤器。如上面所注意到的，由一个过滤器的源（或目的）前
缀和另一个过滤器的目的（或源）前缀创建一些区域可以与部分指明的过滤器聚合（参见图15，区域R5、R6和R7），并且针对这样的区域的项不需要被添加到主表中。

在图16C中示出主表1630的实施方案。参照此图，主表1630包括多个项1632，每个项包括关键字1637a以及一个或更多个容器指针1637b。由在并行LPM搜索（见图16B）中识别的第一和第二索引1691、1692（I和J2）创建的关键字1690被用来搜索主表1630。如果关键字1690匹配主表的任意项1632的关键字1637a，则访问此项中的容器指针1637b标识的容器，以寻找要应用于接收的分组的动作（例如最高优先级动作），下面将更详细的描述该过程。项1632可以指向一个小容器1670，或者可替换地，指向小容器组成的组——即包括对应于过滤器交叉的小容器的并集的“大容器”1680。注意到，在一些实施方案中，不是必须储存大容器1680，因为主表的对应项包括指向被包括在大容器中的所有小容器的指针，这发生在例如规则集是由占用连续优先级级别的规则组成的情况下。但是，在其他实施方案中，可以储存大容器，如图16C中所描绘的，多个项1632可以指向或共享相同的小容器（或多个小容器）1670，这是传输层共享的结果。

次表1640a、1640b的每一个都与主表1630类似。但是，用于访问次表中的一个的关键字包括第三索引1693（I3），并且用于访问另一个次表的关键字包括第四索引1694（I4）。如果对主表1630的查询返回匹配则忽略次表1640a-b，并且主表的匹配项与最具体的匹配过滤器对应。但是，如果在主表中没有寻找到匹配，则对次表1640a-b中的一个的查询可以返回匹配，并且该匹配项与最具体的匹配过滤器对应。如果对主表和次表1630、1640a-b的查询没有返回匹配，则将与整个两维过滤器空间对应的默认过滤器（即“*，*”）用作最具体的过滤器。

在一个实施方案中，主表1630和次表1640a、1640b被实现为哈希表。在该实施方案中，可以向关键字（即关键字1690，或第三和第四索引1693、1694）应用哈希函数来创建用于搜索主和次哈希表的搜索关键字。当然，应该理解，哈希表只代表可以实现主和次表1630、1640a-b的一种实施的方式，且可以使用其他替代的数据结构。

我们现在将注意力转向下一个数据结构1700。在图17中示出第二级数据结构1700的实施方案。第二级数据结构1700的构建是由下面讨论的概念指导的，首先，并且最重要的传输层共享允许第一级数据结构中多项共享各种传输层字段集组成的组，并且更确切地，共享三元组集的组，其中每个三元组包括一个或更多个传输层字段、动作、以及相对优先级（即与规则的绝对优先级相区别的规则集中的相对优先级）。因此，每个三元组集的组——即每个小容器——需要被存储一次。其次，尽管与过滤器交叉相关联的两个或更多个小容器的并集在逻辑上被认为是大容器，但是大容器不需要被存储，因为第一级数据结构1600的主表和次表1630、1640a-b包括指向与一个项相关联的所有小容器的指针。再一次，在另一个实施方案中，可以存储大容器。通过传输层共享，第二级数据结构1700简化为被组织成小容器的三元组列表，并且由于传输层共享所提供的项数减少，相比于其他分类技术，第二级数据结构1700的存储器储存要求小。
参照图 17，第二级数据结构 1700 包括多个三元组 1710，每个三元组 1710 包括一个或更多个传输层（或其他）字段 1719a、动作 1719b 及相对优先级 1719c。每个三元组 1710 与小容器 1720 相关联（即小容器 1720a、1720b、…、1720k 中的一个）。小容器 1720a-k 包括三元组 1710 组成的组，并且小容器存储所在的存储器位置可以通过对应的指针 1725 来识别（即通过对应的指针 1725a 来识别小容器 1720a，通过对应的指针 1725b 来识别小容器 1720b，等等）。如之前所述，容器指针 1725a-k 被存储在转发表 1602 中（参见图 16A）。当在对第一级数据结构 1600 查询中识别出与小容器 1720（或如果存储了大容器，则为大容器）相关联的指针 1725 时，将对应的小容器 1720 中的所有三元组与接收的分组进行比较。如果查找不到至少一个匹配，则将与匹配三元组相关联的动作用于接收的分组。在一个实施方案中，该动作具有的优先级为已经经历过的最高优先级。

在一个实施方案中，第二级数据结构 1700 以内容可寻址存储器（CAM）实现，例如三重 CAM。在实施方案中，CAM 可以包括多个项，每个项与第二级数据结构 1700 的三元组 1710 中的一个相关联。在进一步的第二级数据结构 1700 以 CAM 实现的实施方案中，可以并行的搜索 CAM 的多个项（例如与一个或更多个小容器 1720 相关联的三元组 1730）。

下面参照图 18，示出分类分组的方法的实施方案，所述方法可以通过使用上面图 1-17 以及相关文字所显示的两级分组分类器来执行。参照图中的框 1805，接收到分组。如框 1810a 和 1810b 所阐述的，可以在源地址和目的地址上执行查找（参见图 14 的条目 1410、1412 以及图 16A 的条目 1610、1620）。在一个实施方案中，如图 18 所示，可以并行的执行对源地址和目的地址的查询。但是应该理解，这些查询可以不被并行地执行。基于源地址查询识别出与完全指向的过滤器相关联的最长前缀匹配（LPM）以及与部分指向的过滤器相关联的 LPM，如框 1815a 所示。返回与匹配完全指向的过滤器相关联的索引 I1 以及与匹配部分指向的过滤器相关联的索引 I3（参见图 16B 条目 1691、1693）。

然后组合（例如级联）与匹配完全指向的过滤器相关联的两个索引 I1 和 I2，以形成关键字（参见图 16B 条目 1690），并且关键字用于搜索主表（参见图 16A 和 16C 条目 1630），如框 1820a 所阐述。参照框 1820b 和 1820c，索引 I3 被用于搜索次表中的一个，而索引 I4 被用于搜索另一个次表（参见图 16A 条目 1640a、1640b）。再一次，主表 1630 包括所有完全指向的过滤器和完全指向的过滤器交叉，以及不能与其他过滤器聚合的所有不存在的过滤器（即指示器过滤器）。次表 1640a、1640b 中的一个包括在源地址中具有通配符的过滤器，并且另一个次表包括在目的地址中具有通配符的过滤器。但是次表中不包括指示器过滤器。在一个实施方案中，并行地执行对主表和次表 1630、1640a-b 的搜索，如图 18 所示。但是，在其他实施方案中，可以不并行地执行这些搜索。
参照框 1825，将关键字（由 I1 和 I2 形成）与主表 1630 的每个项 1632 中的关键字 1637a（见图 16C）进行比较。如果寻找到匹配，则访问匹配项中的容器指针 1637b，如框 1830 中所阐述。可以以多种方式使用 I1 和 I2 形成的关键字搜索主表。例如，在一个实施方案中，向关键字应用哈希函数并且使用被实现为哈希表的哈希来搜索主表 1630。

当主表 1630 中寻找到匹配时，忽略次表 1640a-b。如果在主表中没有寻找到匹配，则在次表中的一个中寻找匹配——参见框 1835——访问该次表的匹配项中的容器指针，这在框 1840 中示出。注意到次表中只有一个具有匹配项（如果在次表中确寻找到匹配）。但是如果在主表和次表中任一个中都没有寻找到匹配，则使用于整个两维地址空间对应的默认项，并且访问该项的相关联的容器指针，如框 1845 所阐述。在此交界点已经基于分组的网络地址对接收的分组进行了分类，并且过程转移到分类的第二级。

现在参照框 1850，访问由容器指针中的一个标识的小容器（或者在一些实施方案中大容器）。将接收的分组的传输层字段与所访问的容器（参见图 17）的每个项进行比较，以确定在所访问的容器中是否存在匹配项，如框 1855 中所阐述。如果所访问的容器包括匹配项——参见框 1860——返回与该匹配项相关联的动作，如框 1865 中所阐述。参照框 1870，随后可以将所返回的动作应用于所述分组。

如上所述，主表或次表中的一个中项可以包括多个指针，每个指针标识小容器（换而言之即该项与大容器相关联）。因此，在考虑了所访问的容器中所有项之后（参见框 1855），如果还没有识别匹配项（参见框 1860），则过程将考虑其他还没有被考虑过的容器指针（或容器）（参见框 1850）。如果存在要查询的额外的容器，则重复上述用于访问小容器的过程（即框 1850、1855 和 1860）。因此，只要剩余有还没有被访问过的容器指针（则重复用于访问和搜索小容器的过程直到寻找到匹配）。

现在返回图 15，如之前所描述的，诸如 R2 的不能与其他过滤器聚合的区域——即“指示器过滤器”——被添加到主表 1630（尽管没有一个被放置在次表 1640a-b 中）。完全指示的过滤器或过滤器交叉可以包括多个指示器过滤器。这在图 19 中示出，图 19 示出完全指示的过滤器（或过滤器交叉）1910 以及多个其他的“较窄的”完全指示的过滤器（或过滤器交叉）1920。将一个完全指示的窄过滤器 1920 的源地址与另一个窄过滤器 1920 的目的地址组合——再一次，如上所述，由于对目的地址和源地址查找表的查询是独立执行而导致的结果——生成多个被完全包括在过滤器 1910 中并且因此不能与其他过滤器聚合的不存在的过滤器 1930。因此，在一个实施方案中，将这些指示器过滤器 1930 放置在主表 1630 中。但是，在另一个实施方案中，对与任何给定的完全指示的过滤器相关联的指示器过滤器的数量设置上限。对于任何超过指示器过滤器的该限定上限的完全指示的过滤器或过滤器交叉来说，不将该过滤器或其相关联的指示器过滤器放置在主表 1630 中。超过指定上限的过滤器，例如图 19 中的完全指示的过滤器 1910（包括相对多的指示器过滤器），可以被称为“宽”过滤器，因为这样的过滤器通常跨越宽的源地址和目的地址范围。
在图20A-20C中，进一步示出上述的实施方案。应该注意到图20A、20B和20C分别类似于图16A、16B和18。在图20A-20C的每一个中类似的部件具有相同的数字指示。上面对部件的描述在以下文字中对于某些部件不再重复。

首先参照图20A，超过指定的指示过滤器界限的那些过滤器——即宽过滤器被放置在单独的数据结构中，在本文中所述单独的数据结构被称为宽过滤器表1650。宽过滤器表包括针对每个宽过滤器的项。在一个实施方案中，宽过滤器表1650包括类似于交叉组合表（参见图15和19）的数据结构。对于大多数实际的分类数据库来说，希望宽过滤器的数量小。但是，因为宽过滤器可以潜在地在主表1630中创建大量项，所以将这些宽过滤器移动到单独的数据结构（可以用主表或次表并行的搜索）以显著地提高分类过程的效率。

接下来参照图20B，在分类的第一级中，如之前所描述的，从对源和目的地址表1610、1620的查询分别返回四个不同的索引1691、1692、1693、1694（I1到I4）。第一和第二索引1691、1692对应于与完全指明的过滤器相关联的匹配源和目的前缀，而第三和第四索引1693、1694对应于与部分指明的过滤器相关联的匹配源和目的前缀，如上面也描述过的，在考虑宽过滤器的本实施方案中，对源和目的地址表1610、1620的查询还返回第五索引1695（I5）和第六索引1696（I6）。第五索引1695对应于与宽过滤器相关联的匹配前缀，并且第六索引1696对应于与宽过滤器相关联的匹配目的前缀。注意，在源地址表1610和目的地址表1620的每一个中过滤器类型指示1614b、1624b现在表示该项是否与完全指明的过滤器、部分指明的过滤器、宽过滤器相关联。然后，组合第五索引1695和第六索引1696（I5和I6），已形成用于查询宽过滤器表1650的关键字2090，如下面所描述的。

图20C中示出用于两级分组分类的方法1800的另一个实施方案。图20C中所示的实施方案以大部分与之前参照图18描述的实施方案相同的方式进行，但是，图20C中所示的方法考虑了宽过滤器并且该实施方案使用宽过滤器表1650和以及第五和第六宽过滤器索引1695、1696，如上所述。

参照图20C，接收到分组（参见框1805），并且如框1810a、1810b阐述的，在源地址和目的地址上执行查找。基于源地址查询返回与完全指明的过滤器相关联的最长前缀匹配（LPM）、与部分指明的过滤器相关联的LPM、与宽过滤器相关联的LPM（参见图20B条目1691、1693、1695），如框1815a所示。再一次，I1与匹配完全指明的过滤器相关联，I3与匹配部分指明的过滤器相关联、I5与匹配宽过滤器相关联。类似地，根据对目的地址的查询，识别出与完全指明的过滤器相关联的最长LPM、与部分指明的过滤器相关联的LPM、与宽过滤器相关联的LPM——参见框1815b——并返回与匹配完全指明的过滤器相关联的索引I2、与匹配部分指明的过滤器相关联的索引I4以及与匹配宽过滤器相关联的索引I6（参照图20B条目1692、1694、1696）。
如之前所描述的，随后组合与匹配完全指明的过滤器相关联的两个索引 I1 和 I2，以形成用于搜索主表的关键字，并且索引 I3 和 I4 被用于搜索次表，如框 1820a、1820b 和 1820c 所描述的。此外，组合索引 I5 和 I6，以形成关键字（参见图 20B 条目 2090），并且该关键字被用于对宽过滤器表 1650 的查询，这在框 1820d 中阐述。在一个实施方案中，可以并行地执行对主表 1630、次表 1640a-b、以及宽过滤器表 1650 的搜索，如图 20B 中所示。但在其他实施方案中，可以不并行地执行这些搜索。

参照框 1825，将关键字（由 I1 和 I2 形成）与主表 1630 的每个项 1632 中的关键字 1637a（参见图 16C）进行比较，并且如果寻找到匹配，则访问匹配上的中的容器指针 1637b，如框 1830 中所阐述的。但是，在该实施方案中，如果在主表中没有寻找到匹配，则过程转向宽过滤器表——参见框 1885——并且将关键字（由 I5 和 I6 形成）与宽过滤器表中的每个项进行比较。如果在宽过滤器表中寻找到匹配，则访问宽过滤器表的匹配项中的容器指针，如框 1890 中所示。如果在主表和宽表中的任一个中都没有寻找到匹配，并且在次表中的一个中寻找到匹配——参见框 1835——则访问该次表的匹配项中的容器指针，如框 1840 中所阐述的。如果在主表、宽过滤器表或任意表中没有寻找到匹配，则使用对应于整个二维地址空间的默认项，并且访问该项的相关联的容器指针，如框 1845 中所阐述的。如果在主表 1630 中寻找到匹配，则忽略其他表，并且在主表中没有发生匹配，且在宽过滤器表 1650 中寻找到匹配的情况下，忽略次表 1640a-b。基于被接收的分组的网络路径的分类完成，并且过程转移到分类的第二级。注意在图 20C 中未示出分组分类的第二级，因为第二级将之前所描述的方式进行。

现在我们将注意力转向用于创建和/或更新第一和第二级数据结构的方法的实施方案。图 21 显示创建和/或更新用于上面所描述的二级分组分类过程的数据结构的方法 2100 的实施方案。应该注意到，图 21 中所描述的过程包括在分组分类之前（或者在一个实施方案中，在分组分类过程中）执行的预处理操作集。但是，相比于分组分类的处理，所公开的两级分组分类方案所需的创建和/或更新数据结构（例如第一和第二级数据结构 1600、1700）将不那么频繁地被执行。因此，进行所公开的分类技术所需的大量处理被沉重地提前加载（front-loaded）到一系列预处理操作，所述预处理操作需要被相对不频繁地执行。

现在参照图 21 中框 2110，分类数据库中的规则（参见图 10）被分组为规则集，其中每个规则集中的规则具有相同的源和目的地址（参见图 11A 和 11B）。在一个实施方案中，规则集的规则占用连贯的优先级级别，而在另一个实施方案中，规则集的规则不占用相邻的优先级级别，如上面所注意到的。与每个规则集相关联的源-目的对被称为过滤器，如上面所注意到的。如框 2120 中所阐述的，与每个过滤器相关联的传输层（或其他）字段集，以及对应的动作和相对优先级被组织成小容器。

参照框 2130，创建要用于对接收的分组的源和目的地址的并行 LPM 查询的源地址查找数据结构和目的地址查找数据结构（参见图 16A 和 16B 条目 1601）。源地址查找数据结构包括针对数据库中每个过滤器的项，针对每个过滤器的项包括源前缀、过滤器类型指
示（例如是否是完全指明的、部分指明的或宽的），和索引。类似地，目的地查找数据结构包括针对每个数据库过滤器的项，其中每个项包括针对该过滤器的目的前缀、过滤器类型指示和索引。

为了使第一级数据结构完整，构建转发表（参见图 16A 条目 1602）。确定过滤器交叉（参见图 12A-12C），并且寻找与每个过滤器交叉相关联的小容器的并集——即大容器（参见图 11B），如框 2140 中所阐述。还确定指示器过滤器，如框 2150 中所阐述。如上面所注意到的，指示器过滤器是那些不能与另一个过滤器聚合的不存在的过滤器（参见图 15 和相关文字）。然后，完全指明的过滤器、完全指明的过滤器的交叉和指示器过滤器被放置在主表中（参见图 16A 和 16C），这在框 2160 中阐述。再一次，在一个实施方案中，完全重叠的过滤器交叉（参见图 13A-13C）不被放置在主表中。参照框 2170，创建次表，一个次表包括形式为“X*，Y*”的部分指明的过滤器，并且另一个次表包括形式为“*，Y*”的部分指明的过滤器。在可替换的实施方案中，宽过滤器——例如那些包含多个超过指定阈值的指示器过滤器的过滤器（参见图 19）——被放置到另一个表中——即宽过滤器表（参见图 20A 条目 1650）—这由框 2190 示出。

参照框 2180，创建第二级数据结构，如之前所描述的。在一个实施方案中，第二级数据结构包括与每个小容器相关联的三元组集（参见图 17）。小容器由被包括在主表和次表中的指针所标识。在该实施方案中，大容器不被储存在第二级数据结构中，而是大容器简单地由指向两个或更多个小容器的指针列表（在主或次表的项中）所标识。在可替换的实施方案中（其中，例如，规则集包括具有非连贯的优先级级别的规则），与大容器相关联的三元组组成的组被存储在存储器中。

在本文中，已经描述了用于使用最具体的过滤器匹配和传输层共享的两级分组分类的方法的实施方案，以及可以被用来实现所述两级分类方案的数据结构的实施方案。本领域普通技术人员将意识到所公开的实施方案的优点。传输层共享可显著地减少要被存储的三元组的数量，这减少了所公开的两级分类方案的存储器存储要求。此外，由于 TLS 所产生的减少的储存要求，第二级数据结构易于用内容可寻址的存储器（或其他硬件结构）实现。此外，将最具体的过滤器匹配包括到分组分类的第一级中，可以减少分组分类所需的存储器访问次数——使用 TLS 也可以。此外，所公开的两级分类技术所需要的大量数据处理被沉重地提前加载到多个预处理操作（例如第一和第二级数据结构的创建），并且所述处理操作被相对不频繁地执行。

前面的详细描述和附图仅仅是说明性而不是限制性的。提供它们主要是为了清楚和全面地理解公开的实施方案，并且其中不用理解不必要的限制。本领域技术人员可以设计对本文中描述的实施方案的很多增加、删除和修改，以及替换的结构，而不偏离所公开的实施方案的精神和所附权利要求书的范围。
图 3（现有技术）
图 4（现有技术）

<table>
<thead>
<tr>
<th>420a</th>
<th>420b</th>
<th>420c</th>
<th>420d</th>
<th>420e</th>
<th>420f</th>
<th>420g</th>
<th>420h</th>
<th>420i</th>
<th>420j</th>
</tr>
</thead>
<tbody>
<tr>
<td>420k</td>
<td>420l</td>
<td>420m</td>
<td>420n</td>
<td>420o</td>
<td>420p</td>
<td>420q</td>
<td>420r</td>
<td>420s</td>
<td>420t</td>
</tr>
</tbody>
</table>

有效载荷

目的端口

目的IP地址

源IP地址

协议

源端口

450

410
图 5A（现有技术）

图 5B（现有技术）

图 5C（现有技术）
图 7

接收分组  710

级 1  720
基于网络路径分类分组

级 2  730
基于其他字段分类分组

将最高优先级动作应用于分组  740
<table>
<thead>
<tr>
<th>源地址</th>
<th>目的地址</th>
<th>协议</th>
<th>源端口</th>
<th>目的端口</th>
<th>动作</th>
<th>绝对优先级</th>
<th>相对优先级</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.128.10.5</td>
<td>172.128.*</td>
<td>TCP</td>
<td>21</td>
<td>*</td>
<td>接受</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>128.128.10.5</td>
<td>172.128.*</td>
<td>TCP</td>
<td>80</td>
<td>*</td>
<td>接受</td>
<td>13</td>
<td>2</td>
</tr>
<tr>
<td>128.128.10.5</td>
<td>172.128.*</td>
<td>TCP</td>
<td>17</td>
<td>*</td>
<td>阻挡</td>
<td>14</td>
<td>3</td>
</tr>
<tr>
<td>128.128.10.7</td>
<td>172.128.*</td>
<td>TCP</td>
<td>21</td>
<td>*</td>
<td>接受</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>128.128.10.7</td>
<td>172.128.*</td>
<td>TCP</td>
<td>80</td>
<td>*</td>
<td>接受</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>128.128.10.7</td>
<td>172.128.*</td>
<td>TCP</td>
<td>17</td>
<td>*</td>
<td>阻挡</td>
<td>17</td>
<td>3</td>
</tr>
</tbody>
</table>

图 8
图 9

接收分组

级1
基于源/目的地址分类分组，以寻找单个最具体的匹配

级2
基于使用传输层共享的传输层字段分类分组

将最高优先级动作应用于分组
图 10

<table>
<thead>
<tr>
<th>源地址</th>
<th>目的地址</th>
<th>传输层字段</th>
<th>动作</th>
<th>优先级</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>源地址</td>
<td>目的地址</td>
<td>传输层字段</td>
<td>动作</td>
<td>优先级</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>源地址</td>
<td>目的地址</td>
<td>传输层字段</td>
<td>动作</td>
<td>优先级</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>源地址</td>
<td>目的地址</td>
<td>传输层字段</td>
<td>动作</td>
<td>优先级</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>传输层字段</td>
<td>相对优先级</td>
<td>动作</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>传输层字段</td>
<td>相对优先级</td>
<td>动作</td>
<td></td>
<td></td>
</tr>
<tr>
<td>传输层字段</td>
<td>相对优先级</td>
<td>动作</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

图 11B
图 15
图 16A
<table>
<thead>
<tr>
<th>传输层字段</th>
<th>动作</th>
<th>优先级</th>
</tr>
</thead>
<tbody>
<tr>
<td>传输层字段</td>
<td>动作</td>
<td>优先级</td>
</tr>
<tr>
<td>传输层字段</td>
<td>动作</td>
<td>优先级</td>
</tr>
<tr>
<td>传输层字段</td>
<td>动作</td>
<td>优先级</td>
</tr>
<tr>
<td>传输层字段</td>
<td>动作</td>
<td>优先级</td>
</tr>
<tr>
<td>传输层字段</td>
<td>动作</td>
<td>优先级</td>
</tr>
</tbody>
</table>

图 17
图18

1805: 接收分组

1810a: 在源地址上执行查找

1810b: 在目的地址上执行查找

1815a: 识别与完全指明的过滤器相关联的LPM以及与部分指明的过滤器相关联的LPM（I1和I3）

1815b: 识别与完全指明的过滤器相关联的LPM以及与部分指明的过滤器相关联的LPM（I2和I4）

1820a: 使用I1和I2搜索主表

1820b: 使用I3搜索次表

1820c: 使用I4搜索次表

1825: 在主表中找到匹配？

1835: 在次表中找到匹配？

1830: 从主表的匹配项中访问容器指针

1835: 从次表的匹配项中访问容器指针

1840: 使用默认项（*,*)和相关联的容器指针

1850: 访问指针所标识的容器

1855: 将分组的传输层字段与所访问的容器的项进行比较

1860: 匹配项？

1865: 返回匹配项的动作

1870: 将动作应用于分组
图 19
图 20A
图 21