(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 104394090 A
(43)申请公布日 2015.03.04

(21)申请号 201410645536.9

(22)申请日 2014.11.14

(71)申请人 北京航空航天大学
地址 100191 北京市海淀区学院路 37 号

(72)发明人 李云春 付容 曹凯

(74)专利代理机构 北京永创新实专利事务所 11121
代理人 李有浩

(51)Int.Cl.
H04L 12/801(2013.01)
H04L 12/863(2013.01)
H04L 12/701(2013.01)

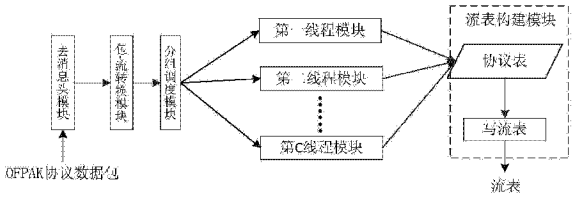
权利要求书2页 说明书8页 附图4页

(54)发明名称

一种采用 DPI 对数据包进行网络流分类的 SDN 控制器

(57)摘要

本发明公开了一种采用 DPI 对数据包进行网络流分类的 SDN 控制器，是在现有 SDN 控制器中增加了采用并行处理方式的 DPI 模块，DPI 模块包括有去消息头模块、包－流转换模块、分组线程调度模块、多个线程模块和流表构建模块；所述流表构建模块中包括有以表格形式存在的协议表和流表。本发明设计的控制器通过修改 OpenFlow 协议实现与网络交换机通信来获取数据包，一方面应用流连接的分组调度，另一方面采用正则匹配将数据包分发给处理线程，最后下发流表到交换机来控制后续数据包的转发。本发明设计的基于 DPI 的 SDN 控制器能够实现 SDN 网络下的较好的 DPI 部署，减少数据包处理速度，提升吞吐量。



CN 104394090 A

1. 一种采用 DPI 对数据包进行网络流分类的 SDN 控制器，是在现有 SDN 控制器中增加了采用并行处理方式的 DPI 模块，其特征在于：DPI 模块包括有去消息头模块、包－流转换模块、分组线程调度模块、多个线程模块和流表构建模块；所述流表构建模块中包括有以表格形式存在的协议表和流表；

去消息头模块用于将接收到的 OFPAK 协议数据包 $OFPAK = \{(head, op_1), (head, op_2), \cdots, (head, op_Z)\}$ 进行去除 OpenFlow 协议头 head，得到原始数据包 $OP = \{op_1, op_2, \cdots, op_Z\}$；

包－流转换模块对接收到的任意一个数据包 op_Z 进行相同五元组内容的拾取，找出所述任意一个数据包 op_Z 对应的流的流连接 ct_B；

分组线程调度模块依据线程权重 qw_C 用于对所述流连接 ct_B 进行处理，得到符合所述 ct_B 的处理线程；

$$qw_C = \frac{LEN_{\min} + flen_B}{len_C + flen_B} g(B,C);$$

多个线程模块从接收到的流连接 ct_B 中提取出数据包 op_Z，然后采用正则表达式方法对所述数据包 op_Z 进行处理，输出所述数据包 op_Z 携带的协议信息 PR 和模式信息 RE；

流表构建模块包括有协议表和流表；所述协议表是将接收到的协议信息 PR 和模式信息 RE 按照协议表形式填入相关项，得到协议结果；然后对协议结果应用策略表得到对应模式名 PA^{CT} 的执行动作 PB^{CT}，最后将执行动作 PB^{CT} 填入流表的指令项中；

写流表是将接收到的协议信息 PR 和模式信息 RE 按照流表形式填入相关项的动作，进而得到流表，然后将流表输出给网络设备。

2. 根据权利要求 1 所述的一种采用 DPI 对数据包进行网络流分类的 SDN 控制器，其特征在于：基于 DPI 的 SDN 控制器对数据包调度的过程有四个步骤；

S1 步骤：支持 OpenFlow 协议的交换机接受到来自网络中设备发送的数据包封装成 OpenFlow 协议数据包记为 $OFPAK = \{(head, op_1), (head, op_2), \cdots, (head, op_Z)\}$，并将 $OFPAK = \{(head, op_1), (head, op_2), \cdots, (head, op_Z)\}$ 发送给基于 DPI 的 SDN 控制器；

S2 步骤：在基于 DPI 的 SDN 控制器中，将 $OFPAK = \{(head, op_1), (head, op_2), \cdots, (head, op_Z)\}$ 中的每个协议数据包的包头去除，实现去消息头的处理，得到 $OP = \{op_1, op_2, \cdots, op_Z\}$；

S3 步骤：在基于 DPI 的 SDN 控制器中，任意一处理线程从运行队列中能够取出流连接 ct_B，得到连接中所有数据包 $OP = \{op_1, op_2, \cdots, op_Z\}$，将数据包 op_Z 应用层数据和系统的规则集 $RE = \{re_1, re_2, \cdots, re_F\}$ 用正则匹配来进行协议检测，得到所有流连接 CT 对应的模式名，将流连接所属协议结果 PR 递送给协议表；

S4 步骤：协议表根据接收到的所有处理线程的协议检测结果 PR，根据协议结果 PR 和系统设定的策略表，得到当前流的执行动作 PB^{CT}，将执行动作 PB^{CT} 填入流表的指令项中，完成写流表，进而得到需要下发给网络设备的流表。

3. 根据权利要求 2 所述的一种采用 DPI 对数据包进行网络流分类的 SDN 控制器，其特征在于步骤 S2 中的关于包－流转换模块和分组线程调度模块具体的步骤如下：

S201 步骤：从步骤 S1 获得原始数据包 op_Z 后，提取数据包 op_Z 的头部五元组信息 srcPort, dstPort, tran, srcIP, dstIP；所述五元组包括源 IP 地址、源端口、目的 IP 地址、目的端

口和传输层协议；然后根据五元组信息找到该数据包 op_Z 信息对应的流连接 ct_B；

S202 步骤：判断流连接表 CT 中是否存在步骤 S201 中生成的流连接标识的条目 ct_B，如果已经存在该流连接条目 ct_B，则转入执行步骤 S203，如果流连接表中不存在该标识流连接条目，转入执行步骤 S204；

S203 步骤：在流连接表中将数据包信息添加到对应流连接条目 ct_B 下，存储数据包信息完成，转入执行步骤 S205；

S204 步骤：在流连接中建立该连接标识的条目，并保存该流连接信息，转入执行步骤 S205；

S205 步骤：获取当前所有处理线程 MT 的任务队列长度 LEN，对每个 mt_C，获取最小任务长度 LEN_{min}，当前 mt_C 的任务队列长度 len_C 和连接 ct_B 的数据包长度信息 $flen_B$，转入执行步骤 S206；

S206 步骤：依据线程权重 $qw_C = \frac{LEN_{\min} + flen_B}{len_C + flen_B} g(B,C)$ 计算当前 mt_C 的权重 qw_C，选择具有最大权重的线程 mt_C，转入执行步骤 S207；

S207 步骤：，将连接 ct_B 加入到具有最大权重的线程 mt_C 的任务队列 qe_C 中，转入执行步骤 S3。

4. 根据权利要求 2 所述的一种采用 DPI 对数据包进行网络流分类的 SDN 控制器，其特征在于步骤 S3 中的关于数据包处理模块具体的协议检测步骤如下：

S301 步骤：处理线程 mt_C 获取其任务队列中的连接 ct_B，得到 mt_C 中所有数据包 $OP = \{op_1, op_2, \cdots, op_Z\}$，执行步骤 302；

S302 步骤：判断 ct_B 的传输层协议 tran 字段是否是属于 TCP、UDP 或者 ICMP，若三者都不是，则丢弃该流连接；若属于其中之一，则进入步骤 S304；

S304 步骤：判断 ct_B 的包的个数 packetnum 是否大于 10，若 packetnum ＞ 10，则丢弃该流连接，若 packetnum ≤ 10，则进入步骤 S306；

S306 步骤：获取数据包 op_Z 的应用层数据进入步骤 S307；

S307 步骤：从规则集 RE 中取一个规则 re_F，将其编译进入步骤 S308；

S308 步骤：将编译后的 re_F 和 op_Z 应用层数据进行正则匹配，若结果为不匹配，则进入步骤 S307，若能够匹配，则进入步骤 S309；

S309 步骤：将协议结果以结果集 $PR = \{pr_1, pr_2, \cdots, pr_B\}$ 形式返回给流表下发模块，并进行流表处理。

5. 根据权利要求 1 至 4 任一项所述的一种采用 DPI 对数据包进行网络流分类的 SDN 控制器，其特征在于：在流表构建模块中策略表是用来约束模式名 PA^{CT} 所对应的流是否转发、丢弃的处理手段，即执行动作 PB^{CT}。

6. 根据权利要求 1 至 4 任一项所述的一种采用 DPI 对数据包进行网络流分类的 SDN 控制器，其特征在于流表的表格格式为：

匹配域	优先级	计数器	指令	超时 定时器	Cookie	标记

一种采用 DPI 对数据包进行网络流分类的 SDN 控制器

技术领域

[0001] 本发明涉及一种 SDN 控制器，更特别地说，是指一种利用深度包检测技术来进行数据包快速分类的 SDN 控制器，特别是基于 SDN 框架下的深度包检测技术的实现方案，并在分组调度和流表下发方面进行优化。

背景技术

[0002] 2013 年 9 月第 1 次印刷，电子工业出版社，《SDN 核心技术剖析和实战指南》雷葆华等编著。在第 15 页图 1-6 公开的 SDN 核心技术体系图中（记为图 1），介绍了在 SDN 架构的每一层次上都具有很多核心技术，其目标是有效地分离控制层面与转发层面，支持逻辑上集中化的统一控制，提供灵活的开发接口等。其中，控制层是整个 SDN 的核心，系统中的南向接口与北向接口也是以它为中心进行命名的。转发层面通过一个 Packet_in 消息将数据包 (Packet，也称为报文）发送给控制层面。SDN(Sofeware Defined Networking，软件定义网络）是一种新兴的基于软件的网络架构及技术，其最大的特点在于具有松耦合的控制平面与数据平面、支持集中化的网络状态控制、实现底层网络设施对上层应用的透明。正如 SDN 的名字所言，它具有灵活的软件编程能力，使得网络的自动化管理和控制能力获得了空前的提升，能够有效地解决当前网络系统所要面临的资源规模扩展受限、组网灵活性差、难以快速满足业务需求等问题。

[0003] 2013 年 10 月北京第 1 次印刷，人民邮电出版社出版发行，《网络流量分类方法与实践》汪立东，钱丽萍主编。在第 116 页中，DPI(Deep Packet Inspection) 深度包检测其概念来自于包检测，之所以称为深度，是由于早期的包检测方法主要检测 IP 包头和 TCP/UDP 包头，而 DPI 方法不仅检测单个数据包的包头，还会对数据包的部分或全部载荷内容进行检测，一般情况下至少要检测超过 64 字节的载荷内容才能够称得上深度包检测，在匹配技术上则要求支持位于载荷中非固定偏移位置起始点的浮动关键词匹配。

[0004] DPI 在 SDN 网络中的位置可能有三种情况：

[0005] (1) 嵌入到应用层：DPI 软件可像其他网络应用一样嵌入到网络应用层，但是这样做深度包检测的瓶颈可能存在于通信路径的长度。因为若要做 DPI，则节点需要将包经过控制器传输然后送到应用层。考虑到延迟因素，这类 DPI 部署方式最好应用于对延时不敏感的应用，如统计分析。

[0006] (2) 嵌入到控制层：DPI 软件可嵌入到 SDN 控制器中，分类信息可用于网络智能化部署也可通过北向 API 传输到应用层以供使用。节点把第一个非空包递交给 SDN 控制器用来做 L4 到 L7 分析。但即使这样，仍有大概不大于 10％的流量需要在 SDN 控制器和 Switch 之间传输才能实现 DPI。

[0007] (3) 嵌入到数据层：网络节点也可运行 DPI 软件，在得到 APP ID 和 metadata（元数据）后可以将其直接应用到预先定义的策略发送给 SDN 控制器和网络应用，并接受返回信息通过 SDN 控制器返回的控制信息，节点做相应 Action（指令），如此相同类型的其他流不需要再做 DPI。这种实现方式延迟最少，但成本最高，因为基于状态机的匹配算法由于其

多模式匹配特性、快速的处理速度、与正则表达式的完美兼容，逐渐成为现在研究最热的匹配算法。研究表明，DPI 性能取决于模式匹配速度。

[0008] 网络流，在一段时间内，一个源 IP 地址和目的 IP 地址之间传输的单向报文流，所有报文具有相同的源端口号 srcPort、目的端口号 dstPort、协议号 tran、源 IP 地址 srcIP 和目的 IP 地址 dstIP，即五元组内容相同。

[0009] 目前设计的 SDN 控制器不具有对网络流进行流量分类，也不能对网络数据包进行控制，因此不能应用于基于流量分类的网络服务。

发明内容

[0010] 为了实现 SDN 控制器对接收到的网络设备输出的数据包进行流分类，本发明设计了一种采用 DPI 架构对数据包进行流分类的 SDN 控制器。

[0011] 本发明的目的是提供一种基于软件定义网络架构的深度包检测技术的连接级并行部署方式，实现对网络数据包进行快速流分类。本发明设计的基于 DPI 的 SDN 控制器是在现有 SDN 控制器中增加了 DPI 模块，所述 DPI 模块采用并行处理方式，即通过修改 OpenFlow 协议，基于 DPI 的 SDN 控制器和网络交换机通信获取数据包，基于连接的分组调度将数据包分发给处理线程，做正则匹配，并下发流表到交换机来控制后续数据包的转发。本发明设计的基于 DPI 的 SDN 控制器能够实现 SDN 网络下的较好的 DPI 部署，减少数据包处理速度，提升吞吐量。

[0012] 本发明设计了一种采用 DPI 对数据包进行网络流分类的 SDN 控制器，是在现有 SDN 控制器中增加了采用并行处理方式的 DPI 模块，所述的 DPI 模块包括有去消息头模块、包－流转换模块、分组线程调度模块、多个线程模块和流表构建模块；所述流表构建模块中包括有以表格形式存在的协议表和流表；

[0013] 去消息头模块用于将接收到的 OFPAK 协议数据包 $OFPAK = \{(head, op_1), (head, op_2), \cdots, (head, op_Z)\}$ 进行去除 OpenFlow 协议头 head，得到原始数据包 $OP = \{op_1, op_2, \cdots, op_Z\}$；

[0014] 包－流转换模块对接收到的任意一个数据包 op_Z 进行相同五元组内容的拾取，找出所述任意一个数据包 op_Z 对应的流的流连接 ct_B；

[0015] 分组线程调度模块依据线程权重 qw_C 用于对所述流连接 ct_B 进行处理，得到符合所述 ct_B 的处理线程；

[0016] $$qw_C = \frac{LEN_{\min} + flen_B}{len_C + flen_B} g(B,C);$$

[0017] 多个线程模块从接收到的流连接 ct_B 中提取出数据包 op_Z，然后采用正则表达式方法对所述数据包 op_Z 进行处理，输出所述数据包 op_Z 携带的协议信息 PR 和模式信息 RE；

[0018] 流表构建模块包括有协议表和流表；所述协议表是将接收到的协议信息 PR 和模式信息 RE 按照协议表形式填入相关项，得到协议结果；然后对协议结果应用策略表得到对应模式名 PA^{CT} 的执行动作 PB^{CT}，最后将执行动作 PB^{CT} 填入流表的指令项中；

[0019] 写流表是将接收到的协议信息 PR 和模式信息 RE 按照流表形式填入相关项的动作，进而得到流表，然后将流表输出给网络设备。

[0020] 本发明的优点：

[0021] ①本发明将DPI部署到SDN架构中的控制层中则流量分类信息可用于网络智能化部署也可通过北向API传输到应用层以供使用。

[0022] ②本发明通过更改OpenFlow协议，使得DPI能够在SDN控制层部署，而无需在各个交换机节点部署DPI，降低成本。

[0023] ③本发明中基于流连接(connection-level)并行DPI方法使得各个处理线程负载均衡，数据流的分组调度更加结合实际流量特点，提高常用规则集的命中率。

[0024] ④在数据包处理模块利用多数据包的多线程同时处理，根据数据流局部性原理调度流，能够更快的处理网络数据包，提高SDN控制器流量分类的处理速度，增大系统吞吐量。

附图说明

[0025] 图1是传统的SDN控制器的体系结构图。

[0026] 图2是本发明的基于DPI的SDN控制器中DPI模块的结构框图。

[0027] 图3是本发明的DPI模块流程图。

[0028] 图4是本发明的包一流转换与分组线程调度的流程图。

[0029] 图5是本发明中流表构建的流程图。

具体实施方式

[0030] 下面将结合附图和实施例对本发明做进一步的详细说明。

[0031] 参见图1所示，本发明是一种采用DPI对数据包进行网络流分类的SDN控制器，该基于DPI的SDN控制器是在现有SDN控制器中增加了DPI模块，所述DPI模块采用并行处理方式，即通过修改OpenFlow协议，基于DPI的SDN控制器和网络交换机通信获取数据包，基于连接的分组调度将数据包分发给处理线程做正则匹配，并下发流表到交换机来控制后续数据包的转发。

[0032] 参见图2所示，在本发明中，DPI模块包括有去消息头模块、包一流转换模块、分组线程调度模块、多个线程模块（第一线程模块、第二线程模块、第C线程模块）和流表构建模块，所述流表构建模块中包括有以表格形式存在的协议表和流表。第一线程模块、第二线程模块和第C线程模块的结构相同。

[0033] 为了更好地理解本发明及其优点，下面结合附图以及具体的示例对本发明做进一步详细的说明。

[0034] （一）去消息头模块

[0035] 去消息头模块用于将接收到的OFPAK协议数据包$OFPAK=\{(head,op_1),(head,op_2),\cdots,(head,op_Z)\}$进行去除OpenFlow协议头head，得到原始数据包$OP=\{op_1,op_2,\cdots,op_Z\}$。

[0036] op_1表示去除了OpenFlow协议头的第一个数据包；

[0037] op_2表示去除了OpenFlow协议头的第二个数据包；

[0038] op_Z表示去除了OpenFlow协议头的最后一个数据包，为了普识性说明，op_Z也称为任意一个数据包，Z表示数据包的标识号。

[0039] 在本发明中，任意一个数据包op_Z包含有源端口号srcPort、目的端口号dstPort、

协议号 tran、源 IP 地址 srcIP 和目的 IP 地址 dstIP 的五元组内容 op_Z = {srcPort, dstPort, tran, srcIP, dstIP}。

[0040] （二）包—流转换模块

[0041] 包—流转换模块对接收到的任意一个数据包 op_Z 进行相同五元组内容的拾取，找出所述任意一个数据包 op_Z 对应的流的流连接 ct_B。

[0042] 在本发明中，SDN 控制器中存在有多个的流连接，所述流连接采用集合形式表达为 $CT = \{ct_1, ct_2, \cdots, ct_B\}$，$ct_1$ 表示 SDN 控制器中的第一条流连接，ct_2 表示 SDN 控制器中的第二条流连接，ct_B 表示 SDN 控制器中的最后一条流连接，为了普识性说明，ct_B 也称为任意一条流连接，B 表示流连接的标识号。所述的任意一条流连接 ct_B 中包含有流连接标识号 ID、数据包的个数 packetnum、流连接的长度 flen、源 IP 地址 srcIP、目的 IP 地址 dstIP、源端口号 srcPort、目的端口号 dstPort 和协议号 tran，采用集合形式表达为 ct_B = {ID, packetnum, flen, srcIP, srcPort, dstIP, dstPort, tran}。

[0043] 在本发明中，SDN 控制器中可能存在多个原始数据包 $OP = \{op_1, op_2, \cdots, op_Z\}$ 对应同一条流连接 ct_B，也可能一个数据包 op_Z 对应一条流连接 ct_B。

[0044] 在本发明中，每一条流连接 ct_B 对应一个流连接的长度 $flen_B$，流连接长度采用集合形式表达为 $FLEN = \{flen_1, flen_2, \cdots, flen_B\}$，$flen_1$ 表示 ct_1 的长度，$flen_2$ 表示 ct_2 的长度，$flen_B$ 表示 ct_B 的长度。

[0045] （三）分组线程调度模块

[0046] 分组线程调度模块用于对任意一条流连接 ct_B 依据线程权重 qw_C 进行处理，得到符合所述 ct_B 的处理线程。

[0047] 在本发明中，$qw_C = \dfrac{LEN_{\min} + flen_B}{len_C + flen_B} g(B,C)$，其中 LEN_{min} 为任务队列长度 $LEN = \{len_1, len_2, \cdots, len_C\}$ 中的最小值，g(B,C) 为固定哈希函数，则 $g(B,C) = \left(a \cdot \left((a \cdot C + b) \oplus B\right) + b\right) \bmod 2^{31}$，常数 a = 1103515245，常数 b = 12345。

[0048] 在本发明中，SDN 控制器包括有多个线程 $MT = \{mt_1, mt_2, \cdots, mt_C\}$，并且每一个线程 mt_C 对应一个任务队列 qe_C，每一个任务队列 qe_C 对应一个任务队列长度 len_C。SDN 控制器中的每一个线程 mt_C 对应一个线程权重 qw_C。

[0049] 线程采用集合形式表达为 $MT = \{mt_1, mt_2, \cdots, mt_C\}$，$mt_1$ 表示第一个处理线程，mt_2 代表第二个处理线程，mt_C 代表最后一个处理线程，为了方便下文说明，mt_C 也称为任意一个处理线程，C 表示处理线程的标识号。

[0050] 任务队列采用集合形式表达为 $QE = \{qe_1, qe_2, \cdots, qe_C\}$，$qe_1$ 表示 mt_1 对应的任务队列，qe_2 表示 mt_2 对应的任务队列，qe_C 表达 mt_C 对应的任务队列。

[0051] 任务队列长度采用集合形式表达为 $LEN = \{len_1, len_2, \cdots, len_C\}$，$len_1$ 表示 qe_1 的长度，len_2 表示 qe_2 的长度，len_C 表示 qe_C 的长度。

[0052] 线程权重采用集合形式表达为 $QW = \{qw_1, qw_2, \cdots, qw_C\}$，$qw_1$ 表示 mt_1 对应的线程权重，qw_2 表示 mt_2 对应的线程权重，qw_C 表达 mt_C 对应的线程权重。

[0053] （四）线程模块

[0054] 线程模块第一方面用于接收流连接 ct_B；

[0055] 线程模块第二方面从流连接 ct_B 中提取出数据包 op_Z；

[0056] 线程模块第三方面采用正则表达式方法对数据包 op_Z 进行处理，输出所述数据包 op_Z 携带的协议信息 PR 和模式信息 RE。

[0057] 在本发明中，正则表达式方法请参考《网络流量分类方法与实践》汪立东，钱丽萍主编，2013 年 10 月第 1 版，第 125-132 页的内容。

[0058] 在本发明中，所有流连接 $CT = \{ct_1, ct_2, \cdots, ct_B\}$ 对应的协议信息记为 $PR = \{pr_1, pr_2, \cdots, pr_B\}$，$pr_1$ 表示 ct_1 的协议信息，pr_2 表示 ct_2 的协议信息，pr_B 表示 ct_B 的协议信息。

[0059] 在本发明中，所有流连接 $CT = \{ct_1, ct_2, \cdots, ct_B\}$ 对应的模式信息记为 $RE = \{re_1, re_2, \cdots, re_F\}$，$re_1$ 表示第一个模式信息，re_2 代表第二个模式信息，re_F 代表最后一个模式信息，为了方便下文说明，re_F 也称为任意一个模式信息，F 表示模式信息的标识号。

[0060] （五）流表构建模块

[0061] 在本发明中，流表构建模块包括有协议表和流表；所述协议表是将接收到的协议信息 PR 和模式信息 RE 按照协议表形式填入相关项，得到协议结果；然后对协议结果应用策略表得到对应模式名 PA^{CT} 的执行动作 PB^{CT}，最后将执行动作 PB^{CT} 填入流表的指令项中。

[0062] 在本发明中，写流表是将接收到的协议信息 PR 和模式信息 RE 按照流表形式填入相关项的动作，进而得到流表，然后将流表输出给网络设备。

[0063] （一）协议结果

[0064]

标识号 ID^{CT}	模式名 PA^{CT}

[0065] 在本发明中，协议结果表示出了哪个流属于哪个模式名（参考《网络流量分类方法与实践》汪立东，钱丽萍主编，2013 年 10 月第 1 版，第 126-132 页的 L7-Filter 模式总结）。

[0066] （二）策略表

[0067]

模式名 PA^{CT}	执行动作 PB^{CT}

[0068] 在本发明中，策略表是用来约束模式名 PA^{CT} 所对应的流是否转发、丢弃的处理手段，即执行动作 PB^{CT}。

[0069] （三）流表的格式如下：

[0070]

匹配域	优先级	计数器	指令	超时定时器	Cookie	标记

[0071] 本发明中引用的流表主体请参考《SDN 核心技术剖析和实战指南》，第 42 页内容，“Cookie”注文为储存在用户本地终端上的数据。不同之处在于：增加了“标记”，所述“标记”是指进入交换机中的流量是否传送到控制器，是一种标记为传送或者不传送的指定。

[0072] 本发明提出的一种采用 DPI 对数据包进行网络流分类的 SDN 控制器，其接收来自

多个交换机（即网络设备）递送的 OpenFlow 数据包，交换机将没有对应流表的数据包作为数据封装在 OpenFlow 协议数据包中，去除 OpenFlow 协议头，得到原始数据包，并对其进行预处理；利用五元组信息将数据包封装为流以建立流连接，若当前流连接为新的，则为其分配空间并将其加入连接队列 CT，并调用数据包调度程序将其分配给系统选定的处理线程 MT，进入 MT 处理队列中。流表构建收集所有 MT 处理结果，对每个流连接根据其处理后的模式名得到关联的策略表，然后利用丢弃、转发等方式对流表中相应的指令字段进行更改，并下发流表到所有交换机。

[0073] 在本发明中，DPI 技术在 SDN 网络架构下具有重要意义。主要表现在以下几个方面：

[0074] (1)SDN 和 DPI 技术结合可以实现集中策略和安全控制。改进的 DPI 技术可以为 SDN 控制器提供网络状态和流量的详细数据。这样 SDN 就可以将网络看作是一个整体的资源，而不是一系列单个设备（如交换机、安全性和其它 4-7 层元素）。DPI 可以为所有相关功能（控制器、策略、安全性等）提供信息帮助，而不是目前各个性能设备的系统各自拥有其专属 DPI 技术。

[0075] (2)DPI 和 SDN 技术结合以提高网络安全性。DPI 技术确保 IT 管理员和安全官员可以制定打击恶意软件和其它威胁的策略，并将其在所有层级实施，包括应用层和用户层。DPI 和 SDN 技术的结合能使网络安全遍布在整个网络，而不仅仅是特定的端点，比如防火墙。

[0076] (3)DPI 和 SDN 技术结合可以在网络管理方面应用大数据。DPI 在为网络健康和性能提供关键信息方面扮演着重要的角色。结合 SDN 的 DPI 技术将引领当前网络走向更容易管理、更安全、运营成本更低的自动化网络。

[0077] 实施例 1

[0078] 以下给出本发明的一个实施例，说明本发明数据包调度的过程（如图 3、图 4、图 5 所示），具体数据包调度步骤如下：

[0079] S1 步骤：支持 OpenFlow 协议的交换机接受到来自网络中设备发送的数据包封装成 OpenFlow 协议数据包记为 $OFPAK = \{(head, op_1), (head, op_2), \cdots, (head, op_Z)\}$，然后将 $OFPAK = \{(head, op_1), (head, op_2), \cdots, (head, op_Z)\}$ 发送给本发明改进的控制器，即基于 DPI 的 SDN 控制器；

[0080] S2 步骤：在基于 DPI 的 SDN 控制器中，将 $OFPAK = \{(head, op_1), (head, op_2), \cdots, (head, op_Z)\}$ 中的每个协议数据包的包头去除，得到 $OP = \{op_1, op_2, \cdots, op_Z\}$；

[0081] 根据任意一个数据包 op_Z 的五元组信息，得到具有相同五元组信息的数据包所属的连接记为 $CT = \{ct_1, ct_2, \cdots, ct_B\}$，且 $B \leqslant Z$，其中 $ct_B = \{ID, packetnum, flen, srcIP, srcPort, dstIP, dstPort, tran\}$；

[0082] ID 表示连接标识号；

[0083] packetnum 表示数据包的个数；

[0084] flen 表示连接的长度；

[0085] srcIP 表示源 IP 地址；

[0086] dstIP 表示目的 IP 地址；

[0087] srcPort 表示源端口号；

[0088] dstPort 表示目的端口号；

[0089] tran 表示传输层协议；

[0090] 根据 ID 将流 CT 分配给数据包处理模块的处理线程

[0091] $MT = \{mt_1, mt_2, \cdots, mt_C\}$，连接 CT 进入 MT 运行队列中，计算对应任务队列 $QE = \{q_1, q_2, \cdots, q_D\}$ 的长度 $LEN = \{len_1, len_2, \cdots, len_E\}$。

[0092] 图 4 中展示步骤 S2 中的关于包一流转换模块和分组线程调度模块具体的步骤如下：

[0093] S201：从步骤 S1 获得原始数据包 op_Z 后，提取数据包 op_Z 的头部五元组信息 srcPort, dstPort, tran, srcIP, dstIP；所述五元组包括源 IP 地址、源端口、目的 IP 地址、目的端口和传输层协议；然后根据五元组信息找到该数据包 op_Z 信息对应的流连接 ct_B；

[0094] S202：判断流连接表 CT 中是否存在步骤 S201 中生成的流连接标识的条目 ct_B，如果已经存在该流连接条目 ct_B，则转入执行步骤 S203，如果流连接表中不存在该标识流连接条目，转入执行步骤 S204；

[0095] S203：在流连接表中将数据包信息添加到对应流连接条目 ct_B 下，存储数据包信息完成，转入执行步骤 S205；

[0096] S204：在流连接中建立该连接标识的条目，并保存该流连接信息，转入执行步骤 S205；

[0097] S205：获取当前所有处理线程 MT 的任务队列长度 LEN，对每个 mt_C，获取最小任务长度 LEN_{min}，当前 mt_C 的任务队列长度 len_C 和连接 ct_B 的数据包长度信息 $flen_B$，转入执行步骤 S206；

[0098] S206：依据线程权重 $qw_C = \frac{LEN_{\min} + flen_B}{len_C + flen_B} g(B,C)$ 计算当前 mt_C 的权重 qw_C，选择具有最大权重的线程 mt_C，转入执行步骤 S207；

[0099] S207：，将连接 ct_B 加入到具有最大权重的线程 mt_C 的任务队列 qe_C 中，转入执行步骤 S3；

[0100] S3 步骤：处理线程 MT 从运行队列中取出连接 ct_B，得到连接中所有数据包 $OP = \{op_1, op_2, \cdots, op_Z\}$，将数据包 op_Z 应用层数据和系统的规则集 $RE = \{re_1, re_2, \cdots, re_F\}$ 用正则匹配来进行协议检测，得到连接 CT 对应的模式名。将连接所属协议结果 PR 递送给流表下发模块。

[0101] 图 5 中展示步骤 S3 中的关于数据包处理模块具体的协议检测步骤如下：

[0102] S301：处理线程 mt_C 获取其任务队列中的连接 ct_B，得到 mt_C 中所有数据包 $OP = \{op_1, op_2, \cdots, op_Z\}$，执行步骤 302；

[0103] S302：判断 ct_B 的传输层协议 tran 字段是否是属于 TCP、UDP 或者 ICMP，若三者都不是，则丢弃该流连接；若属于其中之一，则进入步骤 S304；

[0104] S304：判断 ct_B 的包的个数 packetnum 是否大于 10，若 packetnum > 10，则丢弃该流连接，若 packetnum ≤ 10，则进入步骤 S306；

[0105] S306：获取数据包 op_Z 的应用层数据进入步骤 S307；

[0106] S307：从规则集 RE 中取一个规则 re_F，将其编译进入步骤 S308；

[0107] S308：将编译后的re_F和op_Z应用层数据进行正则匹配，若结果为不匹配，则进入步骤S307，若能够匹配，则进入步骤S309；

[0108] S309：将协议结果以结果集$PR = \{pr_1, pr_2, \cdots, pr_B\}$形式返回给流表下发模块，并进行流表处理。

[0109] S4步骤：流表下发模块收到所有处理线程MT协议检测结果PR，根据协议结果PR和系统设定的策略表，得到当前流的执行动作PB^{CT}，将执行动作PB^{CT}填入流表的指令项中，将1填入流表的标记字段中，并下发流表到所有交换机。

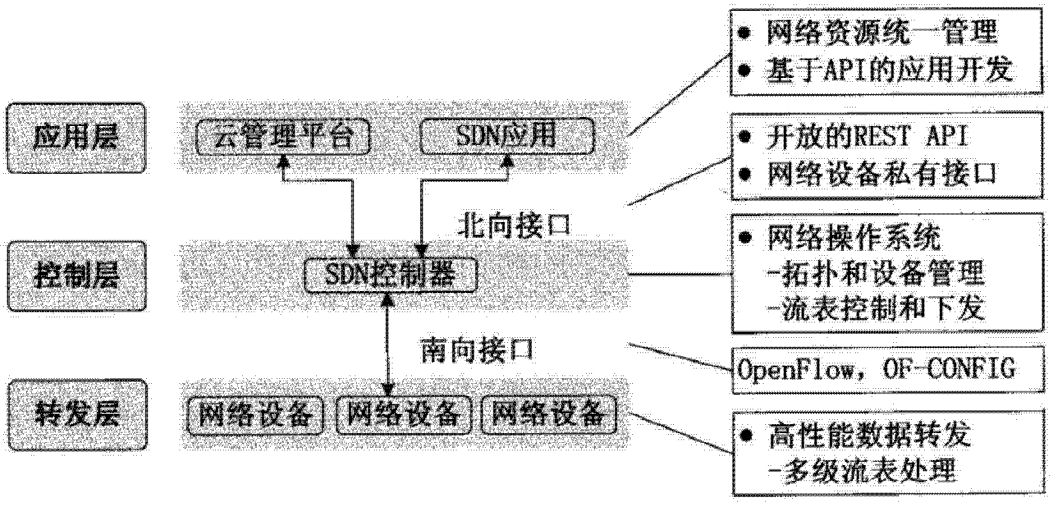

应用层
云管理平台
SDN应用
北向接口
控制层
SDN控制器
南向接口
转发层
网络设备
网络设备
网络设备
• 网络资源统一管理
• 基于API的应用开发
• 开放的REST API
• 网络设备私有接口
• 网络操作系统
-拓扑和设备管理
-流表控制和下发
OpenFlow，OF-CONFIG
• 高性能数据转发
-多级流表处理

图 1

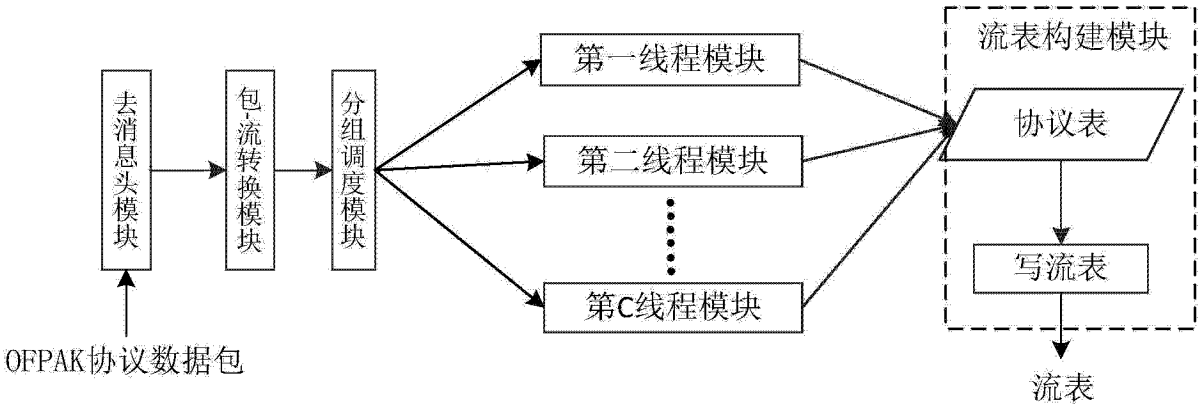

去消息头模块
包流转换模块
分组调度模块
第一线程模块
第二线程模块
第C线程模块
流表构建模块
协议表
写流表
流表
OFPAK协议数据包

图 2

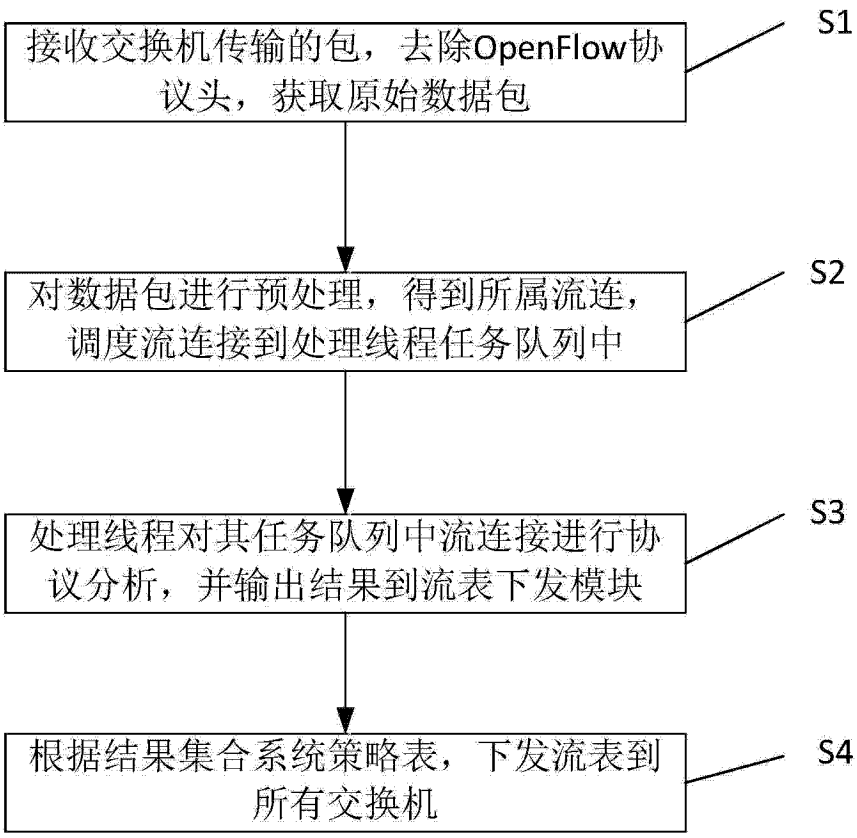

接收交换机传输的包，去除OpenFlow协议头，获取原始数据包
S1
对数据包进行预处理，得到所属流连，调度流连接到处理线程任务队列中
S2
处理线程对其任务队列中流连接进行协议分析，并输出结果到流表下发模块
S3
根据结果集合系统策略表，下发流表到所有交换机
S4

图 3

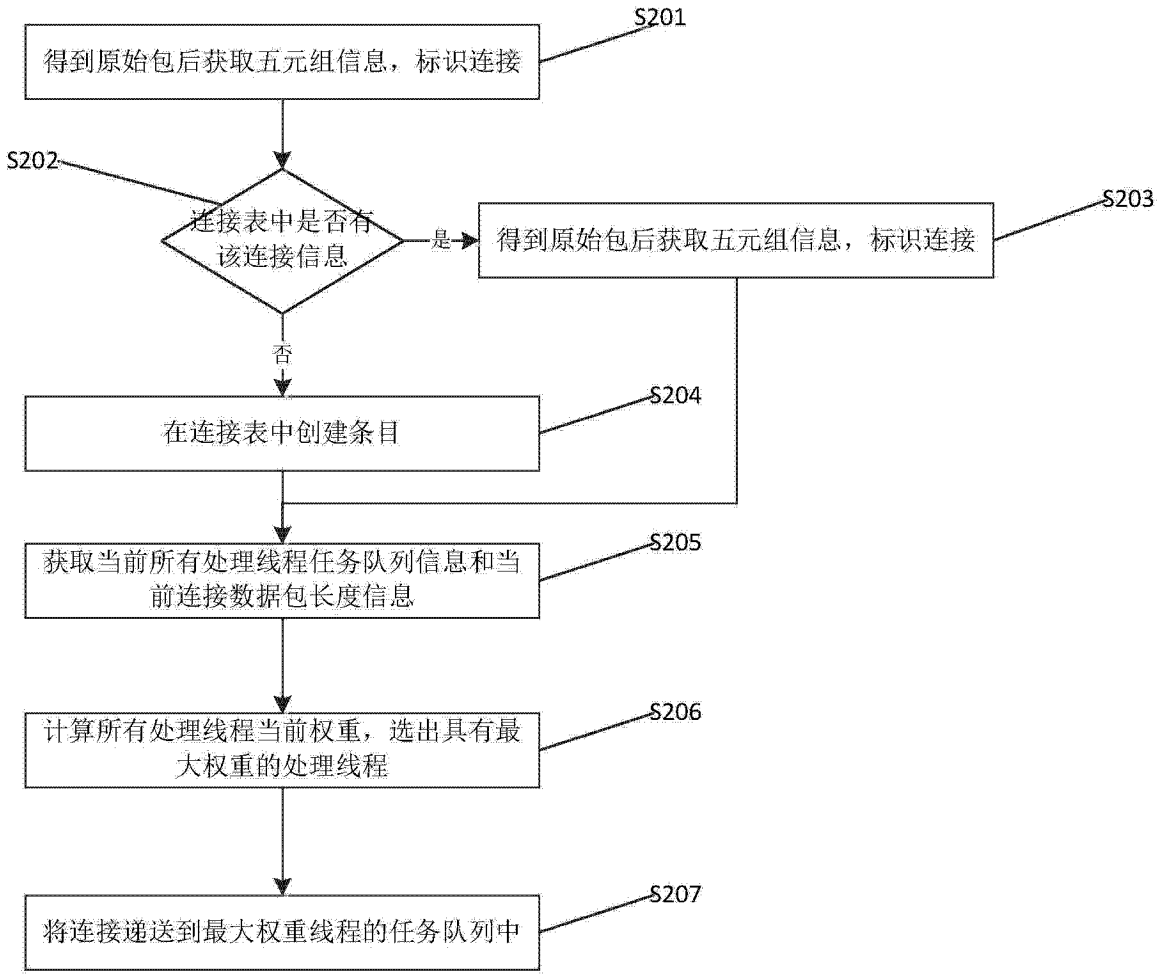

S201
得到原始包后获取五元组信息，标识连接
S202
连接表中是否有该连接信息
是
S203
得到原始包后获取五元组信息，标识连接
否
S204
在连接表中创建条目
S205
获取当前所有处理线程任务队列信息和当前连接数据包长度信息
S206
计算所有处理线程当前权重，选出具有最大权重的处理线程
S207
将连接递送到最大权重线程的任务队列中

图 4

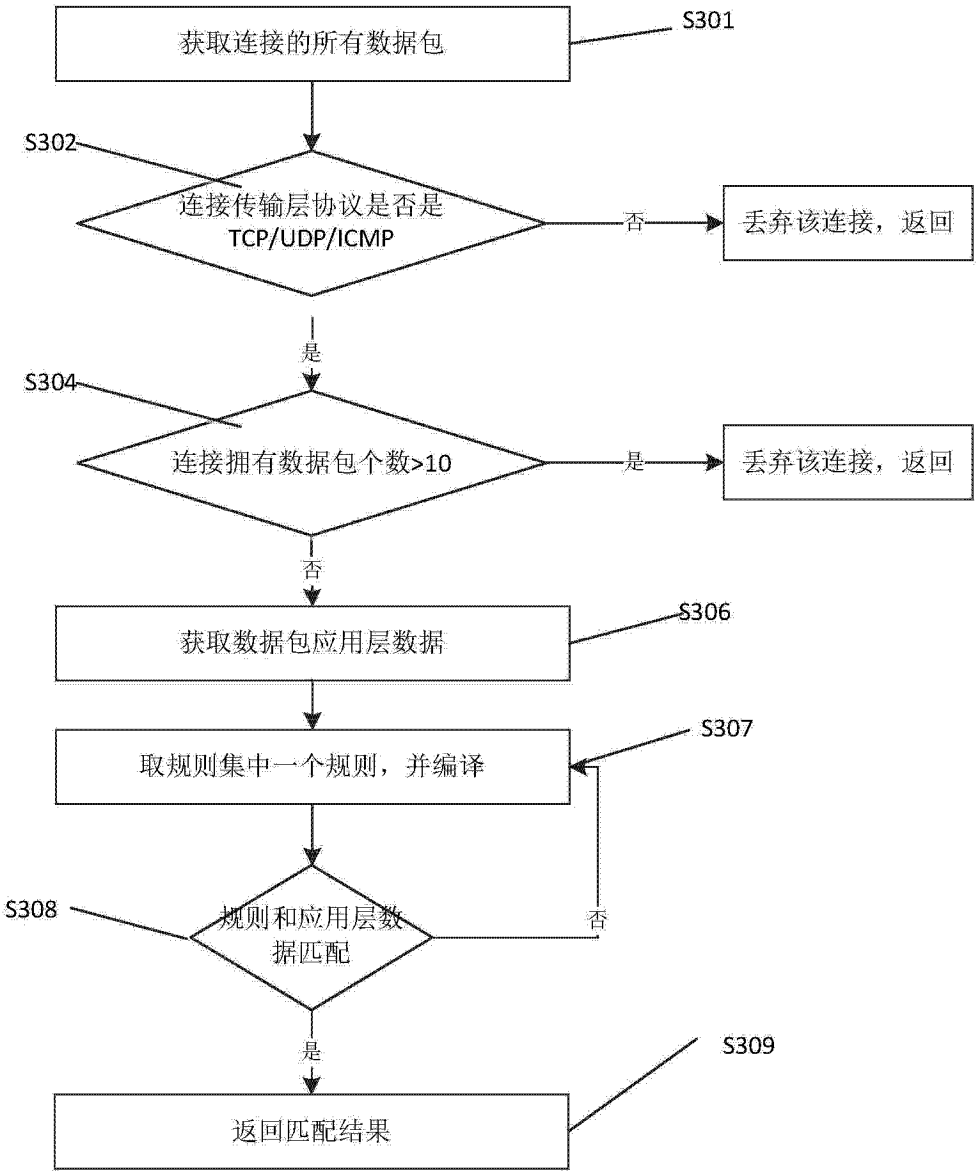

获取连接的所有数据包
S301
S302
连接传输层协议是否是TCP/UDP/ICMP
否
丢弃该连接，返回
是
S304
连接拥有数据包个数>10
是
丢弃该连接，返回
否
获取数据包应用层数据
S306
取规则集中一个规则，并编译
S307
S308
规则和应用层数据匹配
否
是
S309
返回匹配结果

图 5