SDN 交换机转发规则 TCAM 存储优化综述

陈志鹏 徐明伟 杨 芫
(清华大学计算机科学与技术系 北京 100084)
(北京信息科学与技术国家研究中心 北京 100084)

摘 要 软件定义网络(SDN)将传统网络的控制平面和数据平面解耦,通过控制平面的控制器灵活地对网络进行管理,目前应用最广泛的控制协议是 OpenFlow. 三态内容寻址存储器(TCAM)查找速度快、支持三态掩码存储,在 SDN 网络中应用广泛. 但 TCAM 成本高、功耗大,并且在存储含有范围字段匹配域的规则时候存在范围膨胀问题,因此交换机中可存储的转发规则数量,尤其是匹配域的数量和类型都比较多的 OpenFlow 规则数目非常有限,这成为约束 SDN 网络大规模扩展和应用的瓶颈. 研究机构从不同角度提出了针对 SDN 中交换机转发规则的 TCAM 存储优化方案. 本文从转发规则存储架构优化、本地交换机转发规则压缩、全局转发规则动态优化以及控制器参与的网络转发规则管理四个角度总结了相关研究工作,并提出了适合未来 SDN 网络的转发规则存储的综合优化方案.

关键词 软件定义网络(SDN);三态内容寻址存储器(TCAM);转发规则存储优化
中图法分类号 TP393 **DOI 号** 10.11897/SP.J.1016.2021.01341

A Survey on TCAM Storage Optimization for SDN Switch Forwarding Rules

CHEN Zhi-Peng XU Ming-Wei YANG Yuan
(*Department of Computer Science & Technology, Tsinghua University, Beijing* 100084)
(*Beijing National Research Center for Information Science and Technology, Beijing* 100084)

Abstract Software Defined Networking (SDN) makes network control more convenient and more flexible by decoupling the control plane and the data plane. The controller interacts with switches through specific control protocols such as OpenFlow, and rules for forwarding packets will be installed or updated in the switches. Ternary Content Addressable Memory (TCAM) is the most widely used storage medium for forwarding rules in modern SDN switches due to its fast lookup. TCAM supports parallel lookup and outputs the results in one clock cycle. Besides, TCAM can store rules with wildcard directly, which is common in OpenFlow networks. However, TCAM suffers the shortcomings of high cost, high power consumption, and the problem of range expansion when storing range-field-contained forwarding rules. As is known to all, OpenFlow has been the most widely used control protocol at present, and the OpenFlow protocol standard has specified more than 40 match fields according to its latest version. The number of match fields is still growing to achieve high quality network performance. Therefore, forwarding rules in an OpenFlow switch specify longer bits and occupy more room in TCAM than those in a traditional L2 or L3 switch. Therefore, the number of forwarding rules that TCAM can store is very limited, especially the OpenFlow forwarding rules with various match fields. It becomes a bottleneck of software-defined network development. In order to efficiently utilize the limited TCAM storage

收稿日期:2019-10-08;在线发布日期:2020-05-23. 本课题得到国家自然科学基金(61625203,61832013)、国家重点研发计划(2017YFB0801701)资助.**陈志鹏**,博士研究生,主要研究方向为软件定义网络中的转发规则压缩和优化管理机制. E-mail:czp14@mails.tsinghua.edu.cn. **徐明伟**(通信作者),博士,教授,博士生导师,主要研究领域为网络体系结构、高性能路由器、网络安全. E-mail:xmw@cernet.edu.cn. **杨 芫**(通信作者),博士,助理研究员,主要研究方向为网络体系结构、互联网路由器. E-mail:yangyuan_thu@mail.tsinghua.edu.cn.

resources for forwarding rules, it is necessary to optimize the storage problem of forwarding rules in SDN. This paper mainly analyses and summarizes TCAM storage optimization mechanisms of forwarding rules in SDN from such four perspectives as forwarding rule storage architecture optimization, compression of forwarding rules locally and globally, management of forwarding rules with the participation of controller(s). First, forwarding rule storage architecture optimization aims at improving lookup circuit structure in switches, replacing TCAM with better storage media, or applying a mix storage scheme of multiple storage medium. Most methods involve hardware modification. Storing multiple sub-tables rather than the original rule table can improve the storage utilization in TCAM. It is often necessary to maintain information of jump instruction. Second, we can adopt compression algorithms to reduce the number of rules under the premise of keeping rule table semantics unchanged. Compression algorithms need to be effective and efficient, and can compress rule table with high dimension. Third, rule table storage pressure can be relieved by optimizing the routing of flows globally in SDN networks. For example, one rule can be multiplexed by multiple flows in specific network segments, through which the number of rules is reduced. Algorithms are developed in the direction of reducing the number of rules among the entire network. Fourth, Management of forwarding rules with the participation of controller (s) includes caching of forwarding rules or flow table overflow control through specifically designed modules. The module is designed in controllers or additional layer(s) between the control and data plane. However, none of the above four solutions can solve the TCAM storage problem thoroughly. Each of them suffers certain limitations such as overhead of hardware modification, high algorithm complexity, algorithm ineffectiveness and so on. Therefore, we discussed the comprehensive schemes of forwarding rule optimization suitable for future SDN at last.

Keywords Software Defined Networking (SDN); Ternary Content Addressable Memory (TCAM); storage optimization for forwarding rules

1 引 言

文献[1]首次提出软件定义网络 SDN(Software Defined Networking)的概念.广义的软件定义网络是指控制平面和转发平面分离,能够向上层提供资源开放接口、可编程控制的网络架构[2].与传统网络控制和转发紧密耦合的架构相比,SDN 对网络的控制和管理更加灵活.

SDN 对网络数据包的分类和转发有着严格的要求.从程序开发者角度看,许多应用需要实时性;从控制器角度看,控制器需要频繁配置网络中的交换机,因此交换机必须支持快速的规则查找和更新.三态内容寻址存储器 TCAM(Ternary Content Addressable Memory)[3]支持掩码存储,并且与传统的软件包分类算法相比,TCAM 查找速度快、支持并行查找,可以在单个周期内输出所有结果,查询速度与存储的规则数目无关,因此非常适合应用在 SDN 交换机中.但是 TCAM 成本高[4]、功耗大[5],而且当规则中存在范围表示的匹配字段时,会造成存储的范围膨胀问题.例如,当长度为 W 比特的范围区间直接转换为前缀形式的规则存储在 TCAM 中时,最坏情况下规则膨胀系数是 $2W-2$.这极大降低了 TCAM 存储规则的效率.

OpenFlow 是目前 SDN 中应用最为广泛的控制协议,OpenFlow 协议版本 1.4.5 定义的规则匹配域已经增加到 40 多个,而转发规则宽度的持续增加进一步限制了 TCAM 可存储规则的数量.例如,目前商用交换机可存储的 OpenFlow 规则数目在 10^4 数量级[4],而数据中心网络每秒到达的流数量在 100 K[6]数量级上;另外,交换机向控制器请求下发规则到规则下发完成需要的时间约为 50 ms[7],如果新规则到达交换机时 TCAM 规则存储空间已满,需要删除一些旧规则,而这些旧规则对应的数据包到达时需要重新请求控制器下发规则,从而增加数据传输的时延.因此,目前 TCAM 规则存储数量远远不能满足实际网络尤其是数据中心的数据交换需

求，转发规则存储空间成为限制 SDN 尤其是支持 OpenFlow 协议的 SDN 网络大规模应用的瓶颈之一. 如何优化 SDN 网络的转发规则存储，减轻 TCAM 的规则存储压力，成为重要的研究课题.

为了方便讨论，本文统一将交换机中用于分类、转发的所有表项称为规则，其集合相应地称为规则集. 文章根据 TCAM 规则存储资源优化角度的不同，从“转发规则存储结构优化”、“本地转发规则压缩算法”、“网络全局转发规则存储优化”、“控制器参与网络转发规则管理机制”四个角度总结相关工作. 其中，“转发规则存储结构优化”从硬件存储结构角度出发，如优化交换机存储介质和存储结构等，使 SDN 交换机在保证优秀查找性能的前提下，更有效率地使用内部的转发规则存储空间. “本地转发规则压缩算法”在保持原始规则集语义不变的前提下，通过减少规则数目或缩短规则宽度来压缩规则占用的存储空间. OpenFlow 规则维度高、格式复杂、动作指令多样，因此能对 OpenFlow 规则集进行有效压缩的算法少、难度大. “网络全局转发规则优化”则分别根据网络端到端策略、路由转发策略对应的不同规则集的特点，在网络全局范围内平衡转发规则的分布或者减少转发规则的数量. “控制器参与网络转发规则管理机制”则通过在控制器内外设计专门的规则管理模块，通过规则的缓存、超时删除优化和溢出管理等方法，对网络中转发规则的生成、分发和压缩等过程进行管理优化.

本文第 2 节主要介绍如何优化交换机转发规则的存储结构；第 3 节总结本地交换机的规则集压缩算法；第 4 节介绍网络全局范围内通过规则放置、路径聚合、标签路由等方法平衡转发规则的分布或者减少规则的数量；第 5 节总结控制器参与的转发规则管理机制；第 6 节对本文内容作出总结，并提出适合未来 SDN 网络的转发规则存储的综合优化方案.

2　转发规则存储结构优化

SDN 网络要求交换机能够对数据包进行快速转发、转发规则实现快速更新，而传统的软件包分类算法如基于决策树的算法[8-9]、几何区域分割算法[10]、维度分解算法[11-12]、元组空间分割法[13]等，算法复杂度往往较高，难以处理匹配域数量多、种类复杂的 OpenFlow 规则. TCAM 支持三态存储和高速、并行查找，在 SDN 中有着广泛的应用. 但是 TCAM 成本高、功耗大，而且在存储含有范围字段的规则时存在范围膨胀问题，因此 SDN 交换机中 TCAM 可存储的转发规则数目非常有限.

一种比较直接的规则存储优化方案是从硬件角度，对转发规则的存储结构进行优化，主要工作包括 TCAM 自身优化、规则存储芯片优化、混合存储结构和多级规则集存储方案优化.

2.1　TCAM 自身结构

2.1.1　扩展 TCAM(E-TCAM)

文献[14]提出 TCAM 的改造型存储结构 E-TCAM，后者直接存储范围字段，并对范围表示的匹配域进行匹配和查找.

对 16 位的端口范围字段，TCAM 存储相应比特对(lo,hi)，当查询字段 q 到来时，利用设计的对比电路可以直接针对范围字段进行查找匹配. 对比电路分为很多个迭代对比子阶段，每个子阶段代表一个比特位的比较. 例如，在第 i 个比特位，q_i 数值上分别与 hi 和 lo 的相应比特位比较并输出比较结果.

E-TCAM 可以直接对范围表示的匹配域字段进行匹配查找，的确会大大缓解 TCAM 存储范围字段存在的膨胀问题. 但实际情况下由于交换机的产品更新周期长，通常需要数年，而这种从硬件上修改存储结构会增加技术普及的成本和开销，同时针对范围表示的字段专门设计新的匹配电路结构，也会降低交换机处理规则的灵活性.

2.1.2　二进制内容可寻址存储器 BCAM

二进制内容可寻址存储器 BCAM[15]将三态内容存储匹配的问题转化为二进制内容寻址匹配，BCAM 只存储 0 和 1 两种状态，避免了存储通配符“*”所引起的电路复杂度和功耗，因此在电路结构、功耗和成本上相较于 TCAM 有着绝对的优势.

2.1.3　动态可配置 TCAM

文献[16]设计 NAND-NOR TCAM 和级联 TCAM 混合的存储结构，有效降低了存储芯片的功率消耗；文章还设计了如图 1 所示的动态可配置

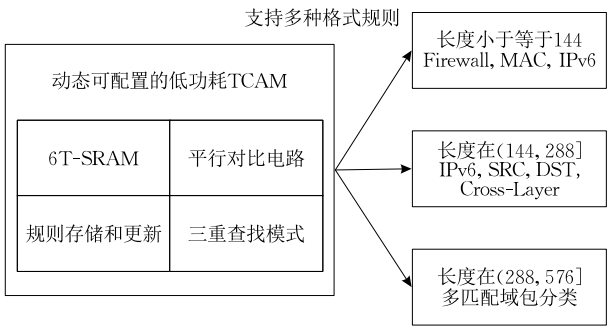

图 1　动态可配置的 TCAM 结构模块

TCAM 存储框架，能够适应性存储不同宽度的流规则，对 OpenFlow 网络来说，该结构有效减小了转发规则存储的时空开销.

2.2 转发规则存储芯片优化

根据 SDN 交换机存储芯片的特点，优化角度可以是 ASIC，交换机 CPU 和 NetFPGA.

2.2.1 专用集成电路 ASIC

SSDP(Split SDN Data Plane)[17]核心系统分为两个模块：传统交换机 ASIC，存储介质为 TCAM，用于存储粗粒度的、存在掩码形式的规则集；子系统由一系列可编程 NPU 单元组成，通过软件存储提供精确匹配的规则集. 两个系统通过 XAUI 接口连接. 如图 2 所示，交换机预先配置为当数据分组到来时，首先经过子系统与控制器通信，再由控制器决定转发规则直接存储在 TCAM 中还是子系统中.

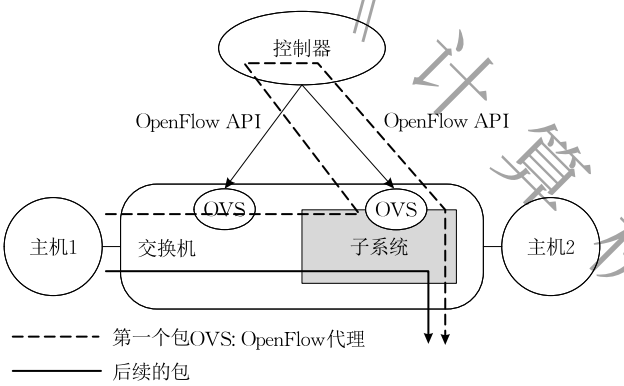

图 2 SSDP 交换机结构

SSDP 交换机通过子系统存储精确匹配的规则，增大了交换机可存储转发规则的数量，从而缓解整个交换机的规则存储压力.

在 OpenFlow 交换机中，计数器为每一条流计数，会占据昂贵的 ASIC 空间，增加电路设计的成本和复杂度. 文献[18]提出去掉 ASIC 中的计数器及相关电路，通过 ASIC 的片上事件记录缓存空间计数，从而使 ASIC 有更多空间存放 TCAM、增加交换机可存储转发规则的数目.

2.2.2 交换机 CPU

DevoFlow[19]认为数据中心里面存在大象流和老鼠流. 其中大象流只占 10%数目的流，却贡献了数据中心网络 90%的流量.

文献[20]充分利用 CPU 的处理能力，为交换机配置更强大的 CPU 作为辅助处理网络流量的协处理器. CPU 和 ASIC 之间通过高带宽的内部链路进行通信. 优化后的交换机结构可以完成 ASIC 处理大象流、CPU 处理老鼠流的分工，缓解 ASIC 上 TCAM 的规则存储和转发压力.

2.2.3 网络现场可编程门阵列 NetFPGA

文献[21]提出了网络现场可编程门阵列 NetFPGA，开发平台成本低、可开源. 转发规则在存储时，NetFPGA 通过片上的 TCAM 和片下的 SRAM 相结合，新的平台不仅支持匹配域中含有通配符“*”的规则快速查找，还增加了平台可存储的 OpenFlow 转发规则数量.

作为创新开放平台，NetFPGA 模块化和可重复使用的特性便于设计者根据网络硬件状态及时修改和拓展网络功能.

2.3 混合存储结构

将需要存储在 TCAM 中的规则集进行分类，结合不同存储介质的成本和功耗等优势，将分类后的规则集在不同存储介质中分别存储，分担 TCAM 存储规则的压力. 图 3 总结了规则集的不同分类算法，以及相对应的存储方案.

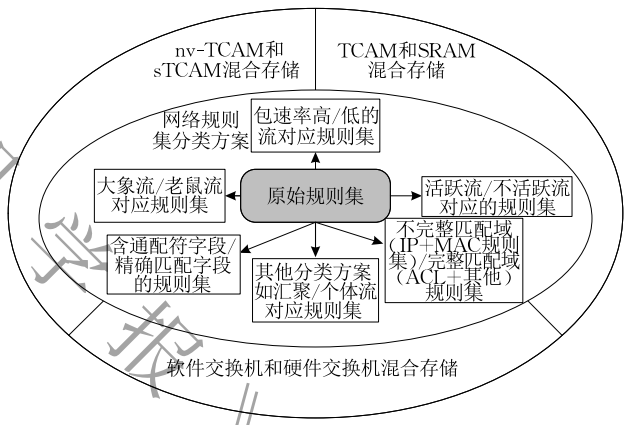

图 3 多级规则集存储结构

2.3.1 TCAM 和 SRAM 混合存储方案

TCAM 规则查找速度快但是成本和功耗高，SRAM 成本和功耗低，但是不支持通配符在任意位置的规则集存储和查找. 如果将两者的优点结合起来，设计 TCAM 和 SRAM 的混合存储方案，可以大大缓解单独使用 TCAM 的规则存储压力，这也是目前混合存储方案里最常见的一种方法.

网络规则集可以分为含有通配符的规则集以及精确匹配的规则集，可以将含有任意通配符的规则集存储在 TCAM 中，需要精确匹配的规则集存储在 SRAM 中. 文献[22]提出基于散列表-TCAM 的混合存储方案，通过特定的散列函数，将原始 OpenFlow 规则中的匹配字段映射压缩为固定长度的指纹，指纹比较用于查找匹配的规则项. 散列表存储在 SRAM 中. 散列冲突时，溢出的条目存储在 TCAM

中. 文献[23]提出了 TCAM 和 Bloom 过滤器混合存储的方案. 方案通过特定的语义转义算法，保证存储在 Bloom 过滤器中的规则在语义上与原始规则的相同. 两种方案的缺点在于提出的减少散列的冲突和 Bloom 过滤器的误判率的方法，会额外引入非常大的计算开销.

针对数据中心网络大象流和老鼠流两种不同流的特点，文献[24]设计了介于控制平面和交换机平面之间的缓存层，通过动态的哈希算法将大象流和老鼠流的信息保存在缓存层 SRAM 中，缓解了由于老鼠流的泛滥而导致的大象流无法获得稳定带宽从而可能被清除的问题. 和传统大象流/老鼠流的分类不一样，文献[25]将流分为活跃流和闲置流，两种流对应的匹配信息分别存储在 TCAM 和 SRAM 中，内容信息存储在 DRAM 中，从而减轻 TCAM 的转发规则存储压力. 在此基础上，根据掩码访问不均匀和访问多次会失败的特性，文献[26]提出针对 SRAM 的加速查找算法，解决了 SRAM 的规则查找性能瓶颈.

文献[27]旨在解决传统 TCAM 需要输出多个匹配规则的包分类问题上的不足，提出利用二元决策树产生 TCAM 兼容的转发规则的索引规则，并将规则信息存储在 SRAM 中，避免了传统 TCAM 在处理多个匹配结果时需额外使用的 TCAM 规则或者字段. 基于 SRAM 的 TCAM(sTCAM)可以在保持低功耗的情况下模仿 TCAM 的功能，但却大大降低规则的更新速度. 文献[28]提出了低延迟的绑定更新(BU-TCAM)方案，在单条和多条规则同时更新的情况下可以有效降低时延.

FEA(Flow Entry Agent)[29]将交换机可存储的规则集分为片(ASIC)上和片下规则集，其中片上规则集包含 MAC 规则集、IP 规则集、ACL 规则集，而片下则通过 SRAM 存放其余的规则集. 片下规则集和 ACL 规则集都含有完整的匹配域，数据包进行匹配转发时，优先与存储在片上的规则进行匹配，找不到匹配的规则时再从存储在 SRAM 中的规则集进行匹配. FEA 通过维护特定的决策表和选择表来记录和选择交换机中分布存储的规则.

2.3.2　nvTCAM 和 sTCAM 混合存储

与传统的由 SRAM 组成的 TCAM(sTCAM)相比，由新型非易失性存储器件 NVM 组成的 TCAM(nvTCAM)能量消耗更低，存储容量更大，但写操作的延迟较高. 文献[30]提出 nvTCAM 和 sTCAM 相结合的混合存储结构，设计了相应的规则迁移算法，并依据规则的依赖关系设计规则清除算法. 混合存储方案使用 nvTCAM 缓存最流行规则，提高缓存命中率；充分结合了 nvTCAM 和 sTCAM 的存储优点，使用少量的 sTCAM 去处理缓存-丢失流量，从而有效减少更新时延. nvTCAM 和 sTCAM 通过规则迁移和替换算法有效结合.

2.3.3　软件和硬件交换机混合存储

NetSoft 算法[31]提出硬件存储和软件存储级联的结构. 顺序上，为更好发挥硬件查找的性能，TCAM 存储放在前面. 新规则到来时，控制器需要对规则在两个存储空间内进行分配. 为了尽可能降低软件存储的使用率，NetSoft 将数据包速率较低的流对应的转发规则转移到软件存储介质中去，而数据包速率较高的流对应的转发规则留在 TCAM 中. 由于算法假定了数据包的速率维持不变，并且在分配规则时候 TCAM 剩余存储空间的计算方法也有待进一步研究，对于更复杂的网络环境，性能无法保证，在实际部署中的表现不明确.

不管是 TCAM 自身优化，还是优化存储芯片或者使用混合存储结构，或多或少会涉及到对交换机硬件的修改，方案部署的开销和成本限制了它们在现有网络上大规模应用.

2.4　多级规则集存储方案

OpenFlow 协议 1.1 版本定义了多级规则集，将需要存储的整张规则集，根据匹配域拆分为一系列的子规则集，中间采用跳转指令连接这些规则集，然后再存储在 TCAM 中，如图 4. 相比单个规则集存储，经过合理构造的多级规则集在存储时更加灵活，可以更有效地利用 TCAM 的规则存储空间.

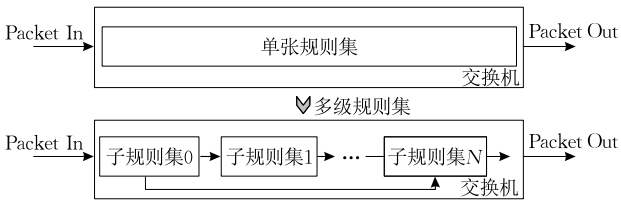

图 4　多级规则集存储结构

2.4.1　根据匹配域逻辑关系构造

H-SOFT[32]根据 OpenFlow 规则的匹配域之间的共存和冲突关系，将具有多个匹配域的单张规则集拆分为一系列的多级规则集(设级数为 k). FTM 模型(Flow Table Management)[33]在此基础上，针对 H-SOFT 得到的 k 个子规则集，进一步进行优化. FTM 将每一个子规则集的匹配域按照线性匹配、精确匹配、最长前缀匹配和范围匹配进一步细

分,并通过专门的CIM(Chip Integration Module)维护这些匹配域之间的跳转映射关系.相比H-SOFT,FTM模块可以更加细粒度的划分子规则集,从而更加灵活地将子规则集在TCAM中进行分配,提高TCAM规则存储的空间有效利用率.

2.4.2 通过高效的映射机制构造

文献[34]提出的多级规则集映射机制,拆分规则集的出发点在于单个匹配域可能会存在多个重复的规则.比如多个规则可能具有相同的IP地址匹配域.算法按照每个匹配域重复规则的数量,计算映射增益从而得到拆分匹配域的顺序.每一次映射拆分,删除相应匹配域的冗余匹配信息,并维护相应的跳转表,直至多次映射完成,得到没有冗余重复匹配信息的多个子规则集.

由于映射过程,删除了每个匹配域内的冗余重复信息,并且子规则集之间的跳转信息由SRAM进行存储,TCAM的有效规则存储空间得到充分有效利用.与原始单一规则集的存储方法相比,文献[21]所提算法可以节省TCAM规则存储容量的17%~95%.算法的局限性在于需要维护额外的规则集用于规则更新;并且交换机规则更新时会涉及到每个规则的掩码修改,算法更新的复杂度较高.

FlowAdapter[35]是一种适应性规则维度转化方案.FlowAdapter综合考虑控制器的需求和开销,以及交换机的异构性,将控制器下发的规则维度(M级),转换为特定交换机硬件支持的规则形式(N级规则集),转换过程保持转发语义不变.表1总结了原始规则集多级拆分的主要方法和局限性.

表1 多级规则集匹配优化方案

算法	思路	依据	局限性
H-SOFT	单个规则集拆分为多个子规则集	匹配域共存和冲突关系	更新需要考虑原规则集表达形式
FTM		匹配域格式	
多级规则集映射		映射增益	
FlowAdapter	交换机适应性规则集维度转换	控制器和交换机支持的规则形式	转化过程依赖原始规则集,更新开销大;交换机处理负担增大

本节从TCAM自身优化、交换机转发规则存储芯片优化、采用混合存储的结构以及多级规则集的存储方案四个角度总结了转发规则存储结构优化的研究内容.其中,前3节从硬件角度扩展了交换机转发规则存储的空间,但是由于交换机的硬件更新周期长,往往需要数年,因此硬件方案的更新和部署成本开销大,无法快速大规模在实际网络中进行应用.而多级规则集存储的方案,可以提高交换机在存储转发规则时TCAM的空间有效利用率,但是算法往往需要维护额外的跳转指令,复杂度较高,会影响交换机更新规则和转发数据包的性能.

3 本地转发规则压缩算法

通过软件方法对交换机中的规则进行压缩,在保持规则集语义不变的情况下,减少规则的数量,或者缩短规则的长度.传统的针对ACL的压缩算法有很多,比如动态规划算法(TCAM-Razor[36] Diplomat算法[37] Lossy Compression[38]等)、二叉树(如MDTC[39])等,往往要求规则集的匹配域是前缀形式表示.但OpenFlow规则的维度更高、匹配域格式更为复杂,大多数软件压缩的方案无法直接应用于OpenFlow规则集.

图5所示OpenFlow规则本地压缩的三个思路,目的都是减少规则集在交换机的TCAM中占用的空间.首先可以在比特层面对规则集做列交换,构造更多压缩机会;虽然通过逻辑运算的方式进行规则集压缩也是以减少规则数目为目的,但是由于后者侧重点在于逻辑运算,因此做了单独归类;最后,我们可以根据匹配域之间逻辑关系缩减冗余匹配域,减少规则的宽度.

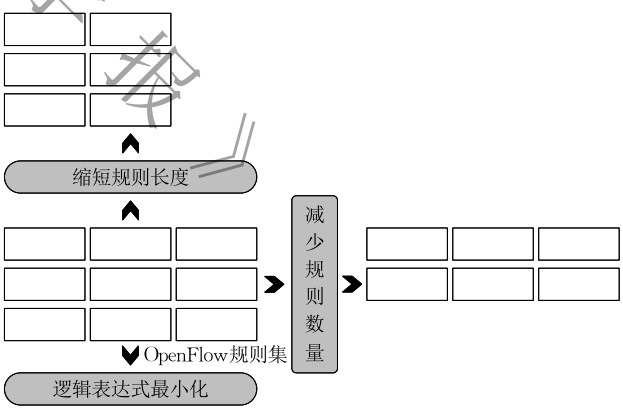

图5 交换机本地规则压缩方案汇总

3.1 压缩规则集中规则的数目

3.1.1 规则集压缩算法

OpenFlow规则中,通配符"*"可以存在于规则中的任何地方,在比特层面通过列交换构造出前缀形式规则,针对前缀表达形式的规则进行压缩.

Bitwaving[40]最早提出针对非前缀形式的规则进行压缩.如图6,Bitwaving自上而下将规则集进行分组,原则是每一个分组的规则集可以通过列交换转换为一维前缀表达形式.组内通过比特列交换

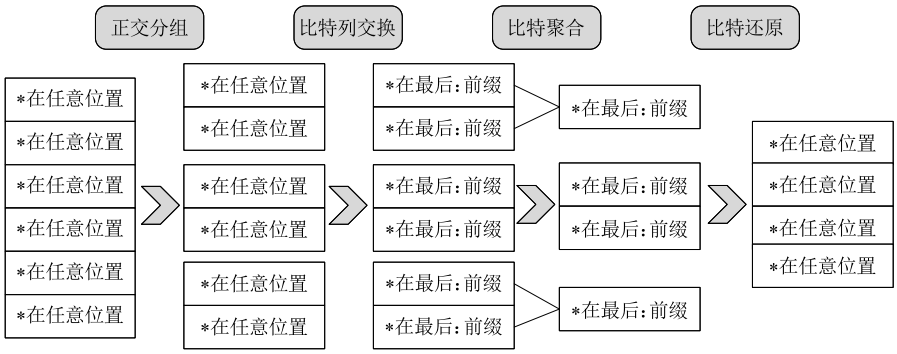

图 6　Bitwaving 算法流程

转换为前缀形式，将汉明距离为 1 的两个规则字符串进行聚合，得到压缩后的新分组规则. 最后再将压缩后的规则集还原为原始比特顺序.

Bitwaving 的压缩效率受到原始规则集“*”的分布限制，分组的数量可能会很多且由于组间列交换顺序各异，列的交换和还原会影响压缩的速度和更新的效率. FFTA(Fast Incremental Flow Table Aggregation)[41] 取消了比特列交换的步骤，对 Bitwaving 得到的分组直接构造二叉树，利用改进 ORTC 和比特聚合方法进行压缩，提高了压缩速度. GenMatcher[42] 将某些原始规则进行膨胀得到前缀表达形式，与剩余规则组成新规则集再进行优化分组，相比 Bitwaving，GenMatcher 的分组更少，查询速度更快. 但由于存在规则数目膨胀，且计算复杂度高，GenMatcher 的配置时间更长. Bit & Subset Weaving[43] 在 Bitwaving 算法得到的分组内，按照规则的动作指令继续划分为一系列的小子集(Subset)，在子集内部和子集之间寻找压缩机会. 文章还提出了压缩算法的触发阈值，可以根据交换机的规则存储资源情况动态执行压缩.

这些压缩算法之间，或者这些算法与一些冗余规则的去除算法如 Redundancy Removal[44] 往往互补，在实际的操作中，可以多种算法协同运行，保证交换机性能的前提下，尽可能多的压缩规则集.

3.1.2　范围字段优化编码

范围表示的字段在 TCAM 中存储存在范围膨胀问题，对于 W 比特表示的整数范围区间，存储为前缀形式的规则时最坏膨胀系数为 $2W-2$，多匹配域的情况下膨胀系数会呈现指数增长. 将范围字段的规则进行编码，用更少的规则表示原始范围字段，可以优化 TCAM 存储效率，缓解范围膨胀. 范围编码的研究工作主要分为两个方向：规则中范围字段数据库依赖和数据库独立，前者编码的有效性依赖于规则集中范围字段的分布情况，而后者则不需要. 图 7 按照上述方向分类总结了相关研究工作.

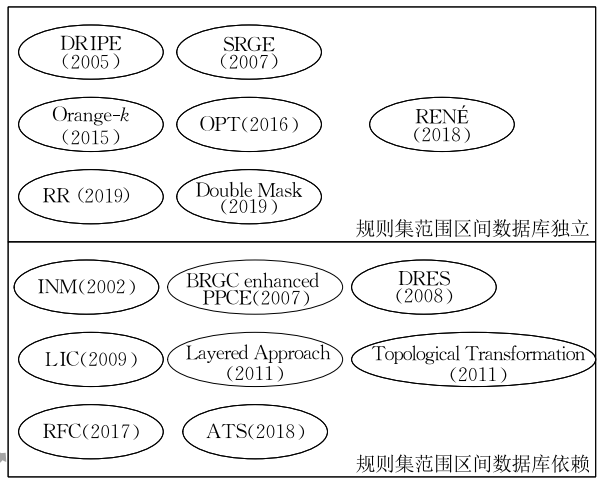

图 7　TCAM 范围字段编码方式分类

并不是所有图中方法都可以直接应用于 OpenFlow 规则集，针对 OpenFlow 规则的范围字段编码方案必须满足支持含有多个范围字段(源、目的端口)、多个动作指令(不止 Accept 和 Deny)的规则集，并支持快速热更新(数据库独立)等.

接下来从三个方面总结适用于 OpenFlow 规则的范围字段优化方法.

(1) 设计范围字段存储和查找结构

Orange-k[45] 在 PIDR[46] 设计结构的基础上，提出了专门为 OpenFlow 规则设计的范围字段编码存储和查找结构. 对于范围区间$[a,b]$，可以根据 a 和 b 的 LCP(最长公共前缀)，计算其 ELCP(扩展最长公共前缀)：0-ECLP 和 1-ELCP. 算法的理论基础是对于任意 $x\in I=[a,b]$，x 的二进制表示至少会与 0-ECLP(a,b)或者 1-ECLP(a,b)中的一个相匹配.

Orange-k 首先对规则集的每一个范围匹配字段进行裁剪，得到范围不相交的、范围字段长度更短的新匹配域，并维护映射树. 多个范围字段的匹配按照图 8 所示结构级联，完成对 k 个范围匹配字段的匹配查找. 可以看出，每个范围字段只需要两条掩码规

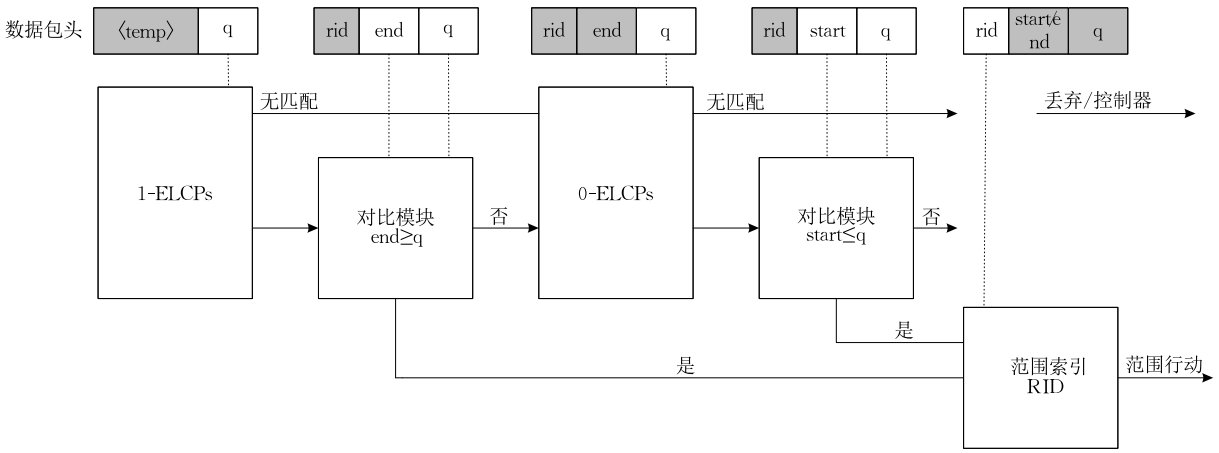

图 8 Orange-1 范围字段匹配结构图

则进行编码存储,大大减少了 TCAM 范围膨胀的问题,同时算法支持规则的快速热更新.但是 Orange-k 交换机结构的修改,部署的开销较大.

(2)利用 TCAM 额外字段编码

RENÉ[47]借助 BRGC[48]编码和 TCAM 中剩余的额外字段,其编码产生的规则数量与最大范围长度成比例,而与范围的个数没有关系. RENÉ将 w 比特长度的范围进行 h 分层:长度为 $w-\log_2(h)+1$ 的 trivial ranges 部分可以直接用单条 BRGC 进行编码,长度为 $h=2^k$ 的 nontrivial ranges 部分无法直接编码,需要辅助 TCAM 长度为 $h-2$ 比特的额外字段进行编码. RENÉ 给出了算法可以编码的最大范围长度和相应需要的额外比特.实验结果显示算法在处理短范围字段的(最大长度 64)时候,需要的 TCAM 额外比特数目更少、更加灵活.

(3)裁剪范围字段长度

RR[49]首先应用 SAX-PAC[50],将原始规则集进行整体分组,得到的每个组的子规则集,范围匹配字段都是长度更短的、无重复冗余的不相交范围(Disjoint Ranges),并维护相应的索引关系. RR 再将这些子规则集按照前缀膨胀或者 SRGE[51]编码的方式转化为 TCAM 可以直接存储的规则表达形式. RR 的理论依据是范围字段越短,转换为前缀或者任意编码形式的字段在进行存储的时候,得到的规则数目就越少,范围膨胀的影响就越小.

除了上述代表性的方案,范围字段优化编码的方案还有很多. Ahmad 等人[52]根据 IP 地址双掩码($netpref/mask1/mask2$,其中 $netpref$ 表示网络地址前缀,$mask1$ 表示可以接受的 IP 地址范围,$mask2$ 表示从包含的 IP 地址字段中需要去除的部分),提出针对 OpenFlow 规则的、对任意范围字段的双掩码表达形式线性时间计算方法.不过算法需要设计专门的交换机电路来存储和处理双掩码形式所表示的规则. RBVE[53]使用了 FPGA,设计了一系列平行分类管道.管道之间整体上是级联形式连接,第一部分是范围字段的存储和匹配阶段,后面是前缀形式的字段匹配.但是 FPGA 的设计成本和开销比较大,并且算法是针对 OpenFlow 协议 1.0 版本定义的规则而设计的.

图 9 总结了针对存在范围字段的规则编码解决方案.鉴于交换机修改成本和开销巨大,支持快速更新的有效编码软件算法成为主要的研究方向.

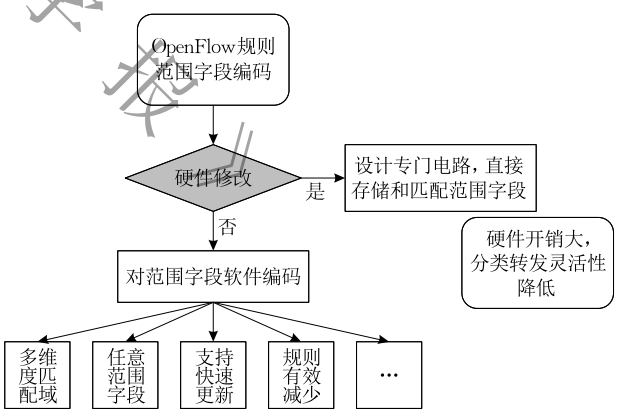

图 9 TCAM 范围字段编码方式分类

3.2 逻辑表达式最小化

逻辑表达式最小化的思路是将原始规则集,看作或者映射成为特定的逻辑表达形式,通过逻辑运算的方式最小化表达式集合.逻辑最小化常用的方法包括卡诺图[54]、Quine-McCluskey (QM)[55-57]算法和 Espresso 算法[58]等.

McGeer 等人[59]提出将原始规则映射到卡诺图,在布尔空间内进行逻辑运算,卡诺图中表现为相邻的、具有相同转发动作指令的空格进行合并,得到

的初步合并后的转发规则再进行压缩优化. BP 算法[60]针对卡诺图中动作指令分布稀疏从而导致压缩效率下降的情况，在 TCAM 前段设计 FPGA 电路，对数据包头进行预处理，提高卡诺图中相同动作指令的分布密集程度，提高压缩效率. 但算法对高维规则的有效性没有验证，且适用的规则集动作指令只限于 Accept 和 Deny 两种，对 OpenFlow 规则适用的有效性无法确定. 此外，BP 算法涉及到 FPGA 电路的设计和对数据分组头的处理，增加了方案的成本，也影响包处理的速度.

文献[61]在构造多级规则集的过程中，迭代进行冗余规则去除，并利用 QM 算法对规则集进行掩码拓展和压缩. 文章还在规则集存储前端设计了基于隐马尔科夫预测模型的 $ExTable$，存放匹配频率和概率比较高的规则，用于优先匹配，降低了匹配时延. 文献[62]将网络中的FIB规则集，利用ORTC[63]得到前缀形式规则集，再简单聚合为非前缀表示的规则集. 将新规则集看作一系列逻辑表达式，运用 Espresso 算法对逻辑表达式进行最小化.

利用逻辑最小化运算虽然能够对规则集进行压缩，但是逻辑运算复杂，规则集的原始状态无法保留，会影响交换机的包匹配处理速度. 虽然文献[59]提出了前置预测匹配模块，降低了匹配时延，但是却相应地增加了电路的设计开销.

3.3　减少规则集中规则的宽度

OpenFlow 规则集匹配域较多，单条规则占用的 TCAM 空间较大，存储效率低；单条规则长度过长也会影响查询的效率. 匹配域修剪是在保持转发语义不变的前提下，调整或者裁剪匹配域，降低单条规则占用的比特宽度.

3.3.1　根据分组头部信息熵裁剪匹配域

Field Trimmer[64]根据“调整规则顺序是否影响匹配结果”这一原则，对原始规则集进行分组：无序集 G_1 和有序集 G_D，前者匹配结果不受规则顺序影响. 对 G_1 规则的每一个匹配域，计算其数据分组头部信息熵，分析其对规则集的区分度贡献. 匹配域裁剪的依据是：数据分组的匹配过程，就是将数据分组头部信息熵降低为 0 的过程. 将规则集图形化表示，计算匹配域组合的信息熵减少量，进而删除掉冗余的匹配域信息.

如图 10，裁剪后的规则集存放 N-TCAM 中，裁减掉的匹配域存放在 RAM 中用于假阳性判定. G_1 和 G_D 的匹配结果输出到优先判定电路，决定最终的匹配结果. Field Trimmer 可以在线性时间内完成对匹配域的提取，但是算法涉及到特殊硬件电路的设计，方案部署的复杂度较高.

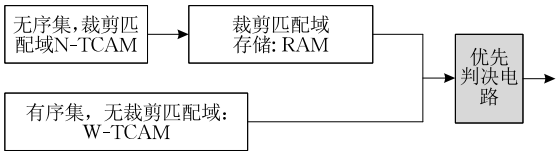

图 10　Field Trimmer 算法

3.3.2　根据逻辑关系裁剪匹配域

在 OpenFlow 规则中，由于各个匹配字段分布的协议层次不同、隶属的协议类型也不同，因而存在共存互斥关系[65]，如图 11 所示.

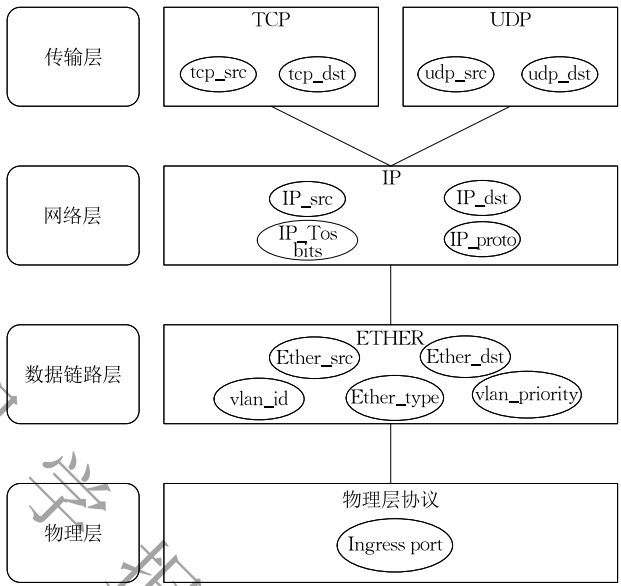

图 11　OpenFlow1.0 包头域分布协议树

在同一层中属于同一节点的匹配字段存在共存关系，在同一层分布不同节点的匹配字段则存在互斥关系. 匹配域之间还有一种演进关系，比如 IPv4 到 IPv6，文献[66]给出了转换具体方案. 可以将匹配域之间的逻辑关系用关系矩阵表示，通过简单的逻辑分析对规则集进行预处理裁剪. 文献[67]对得到的规则集，通过规则独立性进行划分得到各个子规则集分组. 最后对子规则集分组内的规则进行匹配域位提取，再进一步缩短规则的宽度.

对规则匹配域进行裁剪的方案，虽然能缩短规则集的宽度，增加 TCAM 的存储效率，但是大多都涉及到专门的硬件电路设计，不仅增加部署的成本和开销，还会影响规则匹配和更新的效率.

3.4　其他压缩方案

FTRS[68]和 IDFA[69]主要针对单个匹配域的冗余情况进行压缩. FTRS 主要优化局域网络的中间交换机中的规则集，通过选取特殊的匹配域（比如

IP 地址),构造 Trie 树,将改进后的 Trie 树中冗余的节点删除完成压缩的目的.但 FTRS 只适用于局域网的网络环境,且有大概率发生交换机规则溢出.IDFA 压缩依据也是单匹配域如 IP 地址.为了构造压缩机会,IDFA 通过控制器向交换机添加冗余规则,再通过降级、排序等方式对原规则集和冗余规则集进行组合,尽可能多地进行规则压缩.和 FTRS 相比,IDFA 在胖树网络拓扑中有更好的压缩效果以及更短的包传输时延.

TERM[70] 对原始的含有“*”的规则进行整数编码,得到整数表达的新规则集.根据新规则集构造张量矩阵,原始规则集的压缩过程就转换成张量矩阵的降解过程.对 OpenFlow 网络的实验表明,TERM 在减少包等待时间、提高 TCAM 存储规则利用率和降低 Packet_In 信息的频率上有很好的性能.文献[71]在 OpenFlow 协议 1.3.1 版本的基础上,提出了基于资源复用的规则优化存储方案.算法引入了新增的 Mask 值和 Range 值,计算相应匹配域的关联值,进而得到交换机待增规则与已存规则的关联度.增量更新时,通过关联度进行合并或者删除交换机中的规则达到压缩规则集的目的.

本地转发规则压缩主要通过软件算法对交换机中的规则数量进行压缩优化,方案的实施和创新比较方便灵活,但是算法往往很复杂,一定程度上影响交换机的转发规则查找和更新性能;并且有的压缩方案只能适用于特定的规则集,比如针对范围字段的算法只对规则集中的范围字段优化编码和存储,针对特定匹配域的优化方案如 FTRS 等.Bitwaving 虽然可以对任意位置掩码的规则集进行优化压缩,但是其压缩率不稳定,非常依赖原始规则集中通配符“*”的分布.另外,有的算法还需要设计专门的预处理电路,会增加设计成本,引入开销和包处理时延.因此,本地转发规则压缩应该朝着查找和更新速度快、开销小、适用于更多种类匹配域的高效压缩算法方向展开研究.

4 网络全局转发规则优化

SDN 控制器可以获取全局网络信息,可以根据网络拓扑和资源情况,在全网范围内灵活分配转发规则,从而平衡全网转发规则分布;或者通过调整流的路径复用特定转发规则,进而减少全网转发规则的数量.图 12 总结了网络全局转发规则优化方案:从网络策略角度,有分别针对端到端控制策略和路由转发策略的优化算法,也有同时针对两者的联合优化方案;从网络协议角度看,可以是针对 OpenFlow 协议的网络,或者其他协议无关的 SDN 网络.最后,可以借助流量工程,从提高网络服务质量的角度优化全网网络转发规则的存储.

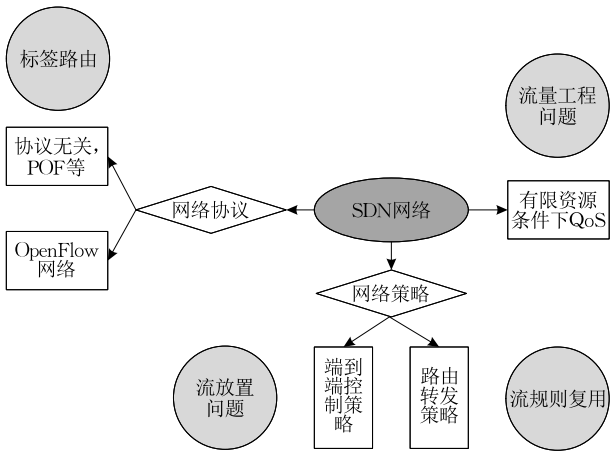

图 12 网络全局转发规则压缩方案

4.1 基于规则放置的优化方案

基于规则放置的优化方案是将整个网络抽象为一个大的交换机,在保持网络的端到端策略语义不变前提下,将相应规则在网络内部特定路径上合理放置,使得网络整体规则数目减少或者均衡分布.规则放置的一般步骤包括规则分割和规则分配,如图 13 所示.

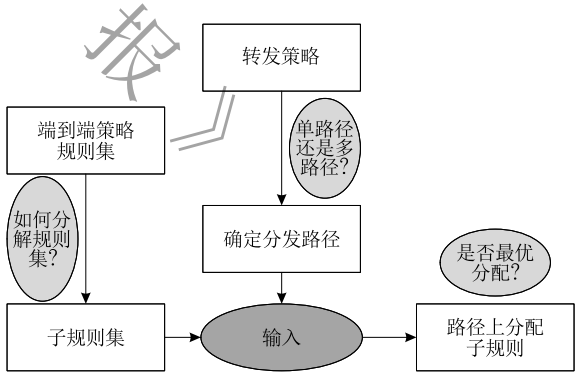

图 13 规则放置算法流程概览

One Big Switch[72] 根据网络的端到端策略和相应的路由转发策略,将网络拓扑分割为一系列的路径集合,并计算路径上单个交换机可分配的规则数目.算法采用贪心策略,在交换机转发规则容量允许范围内,尽可能多的放置规则集.Palette[73] 首先分析规则之间的依赖关系,将原始规则集分割为一系列独立的子规则集.根据端到端策略,计算路径,并通过彩虹着色算法将子规则集在路径上尽可能均衡地分布.One Big Switch 和 Palette 框架更多侧重在

缓解单个交换机资源的存储压力，网络整体的规则数量并没有得到针对性的减少. 文献[74]提出了改进的规则分组和分配方案，相较于 Palette，新的规则分组更小、更加均衡.

CORA[75]从解决交换机规则冲突的角度出发，将交换机中规则分割为相对独立的规则集，然后根据不同交换机之间的规则覆盖关系，沿原始数据包转发路径进行规则迁移和重新放置.

JORA 算法[76]利用规则复用技术，提出多路径的端到端策略规则最小化分布方案，并基于网络约束条件构造混合整数线性规划方程(MILP)，通过软件 CPLEX 求解. 文献[77]同样构造了最小化网络规则的线性规划方程并求解. 在构造线性方程时，除了网络链路、交换机资源限制条件外，作者还分析了交换机规则依赖关系约束条件、路径约束条件. 文献[75]将规则放置的优化方案拓展到多租户数据中心，并且算法利用了网络策略黑名单，在网络入口处就对规则进行了预压缩，结合线性方程的优化目标，算法可以有效减少网络端到端策略规则数目. Raptor[78]的基本思路是尽可能多的让规则被多条路径复用. 首先执行 *Diffuse* 过程，结合规则优先级构造出线性规划方程. 通过 *Diffuse* 过程得到可复用的规则集以及其可以放置的网络拓扑点. 在接下来的 *Connect* 过程中，Raptor 对边缘交换机“对”的规则集构造规则图，分析规则间的依赖关系，并将规则图分区. 最后，参照 *Diffuse* 过程得到的结果对所有分类后的规则集在全网范围内放置.

表 2 总结了端到端策略的规则放置方案，虽然很多算法有效利用或者平衡了网络整体的交换机规则存储资源，但是未对整个网络规则数目进行针对性的优化减少.

表 2　端到端策略规则放置方案汇总

算法	分布方案	核心思路	优点/局限性
DIFANE	垂直分布	权威交换机	有效利用交换机空间
vCRIB		有限	适用于数据中心
Palette	水平分布	规则分组＋规则分布放置	未针对优化规则数目
One Big Switch			未针对优化规则数目
文献[74]		平衡分组	未针对优化规则数目
CORA		规则冲突	未针对优化规则数目
JORA		线性方程	多路径、规则减少
文献[77]			优化减少规则数目
Raptor			优化减少规则数目

4.2　路由转发策略的优化方案

端到端策略将网络看作一个大的交换机，路由策略则关注数据包在网络内部的转发路径. 对路径可调整的流，在网络约束条件下，优化转发路径，新路径下尽可能使网络转发规则数目减少.

4.2.1　路径聚合方法

具有相同目的地的流，如果他们路径发生重合，那么在重合路径上可以通过相同的规则对这几条流进行转发，用于转发这部分流的规则数目“似乎”减少了. 文献[79]根据源地址对流进行分类，对具有相同目的地址的流进行路径分配，分配的原则是在保持链路利用率在较高水平前提下，尽可能多的使得这些路径有重合. 实验结果和基于最短路径的路由策略作对比，在接近最大链路利用率的同时，网络的转发规则得到很大程度减少.

分配路径使尽可能多的流路径重合从而复用某些规则，伴随的问题是可能导致部分流路径拉伸过长，而路径拉伸导致转发跳数增加从而单路径上转发规则数也会相应增加，网络整体的规则数目是否减少还需要进行更深入地算法和实验验证.

4.2.2　求解数学模型

应用图论知识对网络进行建模，在节点交换机规则存储资源、链路带宽等一系列网络状态约束条件下，构造使得网络规则数目最小的线性规划方程组，并求解相应转发路径. 方程是 NP 难的，难以在多项式时间内得到有效求解，因此如何近似求解方程是近年来研究的热点.

OFFICER[80]提出“默认路径＋折射点”的启发式转发策略. 如图 14，算法引入默认路径，进入网络入口交换机 I 的数据包先按照默认路径进行转发，然后在默认路径上的某一点进行转折，通过最短路径到达对应的网络出口交换机 E. OFFICER 采用贪心策略分布需要转发的流，重要的流优先分配路径，从而使尽可能多的流通过网络出口. 文献[81]提出网络转发规则最小化的多路径路由转发方案，根据网络约束条件建立相应的混合线性方程，并提出了基于蚁群路径搜索的 CACO-RSP 算法.

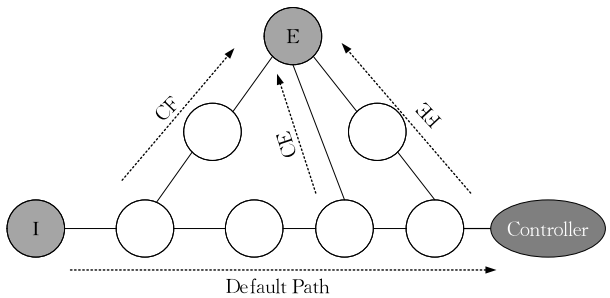

图 14　路由折射点选择策略

MIRA[82]构造混合线性规划模型，优化的目标是动态流量背景下，尽量减少网络转发规则的安装

执行过程. MIRA 采用贪心策略寻找满足方程优化条件的转发路径. 此外,文献还提出了在本地交换机压缩转发规则的方法,进一步减少转发规则安装的频率. FRM[83] 将不同的流通过数据包分组的头 VLAN ID 进行分类标识,使其流经相同或者部分相同的转发路径,复用的路径上多条路径使用相同规则集进行转发. FRM 建立了在 IoT(Internet of Things)场景下,SDN 网络转发规则的最小化数学模型,并通过分布式马尔科夫近似方法对整数线性规划方程近似求解.

图 15 总结了利用数学模型求解网络转发规则最小化的转发路径方案.

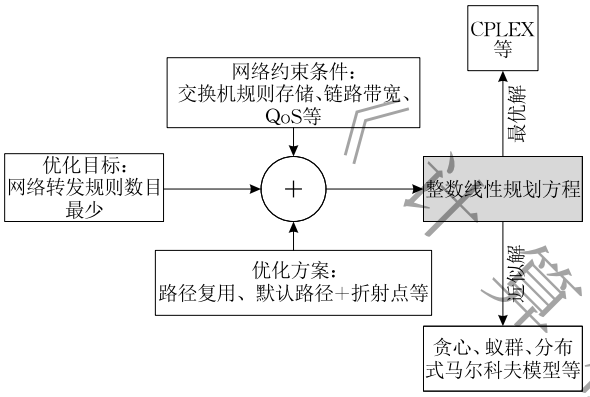

图 15 规则最小化数学模型求解方案

4.3 标签路由算法

文献[84-86]使用 MPLS 标签对 SDN 网络中分组进行转发,数据包在入口交换机压入指示路径的标签,数据在网络中的转发通过 MPLS 的标签进行,避免了 OpenFlow 规则带来的 TCAM 存储压力. 但不可避免的是由于需要转发的数据包携带了转发标签,增加了网络需要转发的无效负荷,降低网络链路带宽资源的有效利用率.

改进的标签路由算法主要集中在两个方面:利用 OpenFlow 自带的协议字段作为转发标签;使用协议无关路由的 SDN 架构;或者优化标签编码,使新的标签占用更少的带宽资源.

4.3.1 利用 OpenFlow 自带的协议字段

JumpFlow 算法[87] 将数据包分组的路由信息写入到分组自带的 VID(VLAN ID)中. 由于 VID 只有 12 个比特位,只能携带 3 跳的路由信息,因此需要将数据包的路径进行分段,其中部分路径的数据转发依据 VID 信息. JumpFlow 通过计算不同分段方案下的交换机规则存储资源利用率,选择最合理的路径分段方式.

由于流的部分路径依据了 VID 进行转发,而非完整的 OpenFlow 规则,因此会缓解 TCAM 存储完整 OpenFlow 规则的压力. 同时 JumpFlow 利用的是数据包分组自带的字段,无需添加新的包头字段,减少了数据包携带冗余信息,提高网络带宽利用率. 文章提出的算法是在线计算,可能会增加控制器的计算负担.

4.3.2 网络不使用 OpenFlow 协议

(1)协议无关路由

POF[88] 是华为在 2013 年提出的一种 SDN 架构,通过重新设定的转发规则三元组(偏移量、长度及动作)来替代 OpenFlow 定义的复杂的协议规则. F-FC[89] 在 POF 环境下,定义了如图 16 所示的转发数据结构.

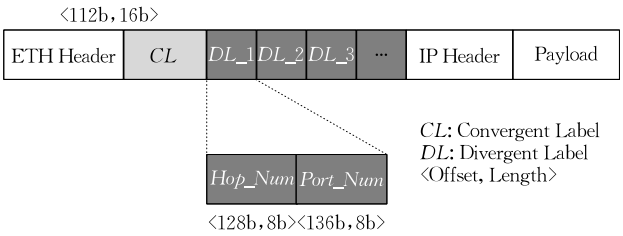

图 16 F-FC 定义的转发数据包结构

网络入口交换机为数据包插入的标签包括 *CL* 和 *DL* 两部分. 其中 *CL*:去往同一个目的地的流聚合到一起,给他们分配一个 *CL* 的标签;*DL*:为了保证流量调度的细粒度,给每条流后面加上一个或者多个的 *DL* 的 label,*DL* 的个数取决于这条流分流的节点. 如图 16,*CL* 的结构形式是〈112,16〉,偏移量是 112 代表 *CL* 直接插入在 *ETH* 字段的后面,16 代表 *CL* 的长度是 16 bit. 每一个 *DL* 都具备两个域,都占有 8 bit:*Hop_Num* 代表从当前节点到分离节点的跳数,*Port_Num* 代表在分离节点上,为包设计的转发端口. F-FC 根据相同目的地的流可能具有聚合节点的思路,构建了流聚合的动态形式化方程,提出了两种启发式算法并做了仿真验证.

POMP[90] 在新型 SDN 网络架构(P4、POF 等)的基础上,提出了整合性的解决方案. POMP 的交换机抽象层,能够分别识别 P4 和 POF 等不同交换机发来的数据包;为了优化 TCAM 规则存储,POMP 利用"污点分析[91]"得到并压缩多级规则集. 相比于 OpenFlow 网络,POMP 能够更加灵活地解析数据包,用于转发的规则集细粒度高、且占用 TCAM 的存储资源少. 除此之外,由于数据包的解析和规则集的生成过程都是自动的,大大减少网络数据传输时延和 Packet_In 信息.

(2)优化标签编码方案

文献[92]在数据中心网络提出优化的标签算法

XPath. XPath 提前计算好“足够多”的静态路径，通过二进制前缀形式为单个交换机内的路径进行随机分配路径 ID；算法的核心在于如何统一各个交换机内的 ID 并进行 ID 聚合压缩；XPath 最后利用数据中心拓扑的对称性等特点压缩这些路径的数量并赋予相应的 ID，最终实现数据中心网络的离线路径计算. 路径 ID 和实际的网络地址映射表存储在控制器中. 数据包分组到达网络中时，向控制器申请相应的路径即可进入网络进行转发. Xpath 算法是基于路径的，转发是粗粒度的；并且其路径编码的压缩比例非常依赖网络拓扑的种类，因此在广域网中的使用会有很大局限性.

PathSets[93] 根据流的各种网络属性进行分组，对包含顺序的等属性子集合分别进行掩码编码压缩. PathSets 的编码方案适用于含网络服务功能链网络，以及基于 SDN 的因特网交换点(IXP). 文献 KeySet[94] 依据中国剩余定理(CRT)，把数据包的路由信息编码进一个标签 L 中，交换机解析数据包时，直接在本地进行模运算，根据 L 的值求解出需要将数据包进行转发的端口. 交换机本地模运算代替查询冗长的规则集，可以在常数时间内完成数据包的转发. 不过算法的实验结果针对数据中心网络，其他网络环境下算法的有效性还有待验证.

表 3 总结了使用标签进行路由的方法思路.

表 3　标签路由算法方案汇总

算法名称	网络类型	标签方案	思路
JumpFlow	OpenFlow 网络	VLAN ID	分段标签
F-FC	POF	〈偏移量，长度〉	流聚合
POMP	POF、P4 等	自定义标签	污点分析
XPath	SDN 数据中心	静态路径 ID	
PathSets	广域网	等属性流集合	压缩编码
KeySet	SDN 数据中心	CRT	

4.4　TCAM 有限条件下的 QoS 指标优化

前面提到的缓解 TCAM 规则存储压力的重要性，有一部分原因是为了提高网络服务质量. 在交换机规则存储资源有限条件下，最大化某个 QoS 指标，意义等效于“为了达到既定目标，增大了 TCAM 的规则承载能力”. 从这个角度出发，可以借助流量工程的方法，在网络交换机 TCAM 规则存储资源、网络带宽资源等约束条件下，最优化某个网络参数指标如最大化网络吞吐量、最大化网络链路利用率和最大化网络能源利用率等.

图 17 总结了利用流量工程提高网络服务质量的步骤. 建立的方程往往是 NP 难的，实际求解方程时需要借助有效的近似算法.

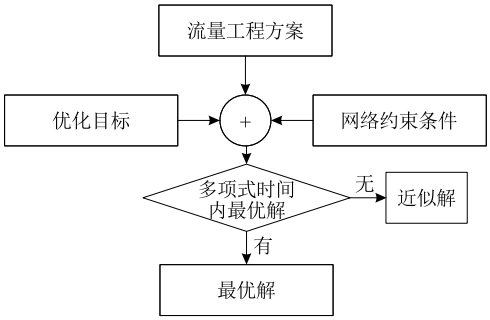

图 17　利用流量工程提高网络服务质量

常见的表示网络质量和性能参数的指标包括吞吐量、能源利用率、负载均衡、链路利用率等. 表 4 总结了相关文献的优化目标和求解方法.

表 4　网络 QoS 优化算法

算法	优化目标	求解方法
文献[95]	吞吐量	连续线性规划
EAR[96]	网络能耗	贪心算法
MINNIE[97]	负载均衡	基于地址的流聚合
文献[98]	联合优化	“三步走”启发算法

文献[95]将整数规划问题转成连续规划，实际的整数规划结果和连续规划结果存在差异，最终的结果通过松弛间隙来评估. EAR[96] 优化的是网络的能源消耗，即网络在运行状态下活跃的链路数量. MINNIE[97] 在已有可选路径前提下(K-Shortest 路径)，对输入网络的流量进行再分配和节点交换机转发规则压缩，在使网络达到负载均衡的状态下尽可能多的压缩网络的转发规则数目. 文献[98]构造整数线性规划模型，对多路径路由和网络转发规则进行联合优化. 文章通过多项式时间内可解的“三步走”启发式算法对方程组进行求解.

优化方案的核心在于如何在多项式时间内有效近似求解方程，或者寻找高效的启发式算法. 与 4.2.2 节求解数学模型的方法不同，QoS 优化的目标往往不是网络流表数目，TCAM 转发规则存储资源仅仅作为约束条件出现在方程组中.

5　控制器参与网络转发规则管理机制

通过 SDN 控制器的参与，对网络转发规则从生成到超时删除等中间各个阶段进行优化管理，分担或者缓解规则向 TCAM 存储的压力. 按照管理阶段的不同，可以分为规则生成和下发管理机制、缓存机制、规则超时和溢出管理机制，如图 18 所示.

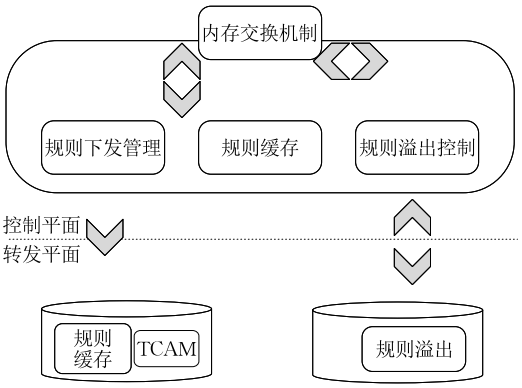

图 18 SDN 网络规则管理模块概览

5.1 规则生成和下发管理机制

规则的生成主要与用户需求和网络环境等因素有关,这里不做赘述. 在转发规则生成后,需要根据网络的 TCAM 资源分布情况进行优化预处理,然后再进行分发,这个过程需要良好的管理机制.

相较传统控制器,Mapple++[99]产生的转发规则更紧凑、有效. Mapple++包含四个模块:核心引擎,负责运行应用程序、记录网络状态等;全局优化模块,根据数据包信息和网络全局信息,做出路由决策;为每个交换机单独配置的本地优化模块,配置合适的规则集,并进行优化压缩;最后一个是环境信息收集模块,收集网络拓扑和设备状态信息. Mapple++下发的规则是多级结构,能够更加有效利用 TCAM 的规则存储空间. 此外,Mapple++的规则下发和更新等都是被动模式,避免了用户主动式更新网络带来的不方便和不确定性.

RL-based Approach[100]通过传统强化学习和深度强化学习算法,对网络中的流量进行学习分类,决定哪些流适合放在交换机的 TCAM 中,哪些流的规则适合放在控制器中. 文献[101]建立并近似求解本地交换机 TCAM 规则占用成本和远程控制器信息处理成本的加权最小化配置数学方程. 按照算法给定路径配置转发规则可以减少交换机规则占用成本和控制器的远程控制信息开销.

TableVisor 2.0[102]提出介于控制平面和转发平面的虚拟管理层. 对控制器而言,数据平面被抽象为一个整体的交换机,"交换机"内部的数据转发通过分布在各个交换机上的子规则集 ID、或者跳转指令进行转发. TableVisor 2.0 将数据平面每一个交换机看作多级子规则集,TableVisor2.0 利用规则优先级和特定匹配域的哈希值将规则集在各个交换机之间进行均衡分布,提高了 TCAM 规则存储资源整体有效利用率. 但是这种网络多层抽象结构的介入,会一定程度上增加网络传输时延.

一些特定场景下的规则下发管理模块,也可以减少网络中的转发规则数目. SAT-FLOW[103]是基于 SDN 的卫星网络管理模块,主要包含两个算法:动态分类超时算法,用于减少交换机中规则数目;超时策略移动管理算法,作用是维持网络移动切换时的丢失流数目. LiteVisor[104]的应用场景是基于 SDN 的网络虚拟化(Network Virtualization),LiteVisor 路由方案将网络虚拟化和标识符分分离协议(Locator-Identifier Separation Protocol,LISP)整合起来,支持流的聚合和虚拟机迁移时无缝网络重构. 与前面通过算法进行交换机规则压缩和网络流聚合等优化方案不同的是,规则下发管理机制往往需要设计单独的管理模块.

5.2 规则缓存机制

控制器生成的规则不是全部直接下发到交换机的 TCAM 中,而是将一部分暂时存放在控制器的 RAM 中,或者交换机专门的缓存模块中.

如图 19 所示的 CAB[105]算法框架. CAB 首先分析规则集空间内的依赖关系,将规则集空间进行分割,并放入合适的缓存桶内. 掩码表示的缓存桶用于数据包的精确转发以及确定哪些规则适合放在缓存桶内. 流建立时,控制器查询相应的缓存桶并将桶内相应规则缓存在交换机上.

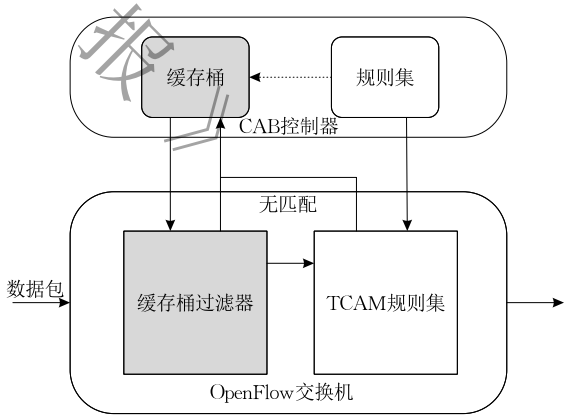

图 19 CAB 缓存框架概览

CacheFlow[106]是一个虚拟的层次结构,一系列的软件交换机作为硬件交换机的代理或者独立的服务器运行在数据平面. CacheMaster 接受控制器消息,并通过 OpenFlow 协议将规则下发给软件和硬件交换机. 因此,CacheFlow 可以把最流行的转发规则集存储在资源有限的交换机 TCAM 里,通过软件处理那些非流行的流信息. CacheFlow 提出基于规则相互依赖关系的分类算法,区分存放在 Cache-

Master 或者交换机 TCAM 中的规则. 当数据包到达交换机时候，先在 TCAM 中查找有无对应的转发规则，如果有就根据匹配字段找到相应的转发动作并执行；如果没有相应的转发规则，就会请求 CacheMaster，在其中找到相应的转发规则，并下发到 TCAM 中，对数据进行匹配转发. 如果都没有找到，就会向控制器发送规则下发请求.

CRAFT[107] 通过分析规则集的空间分布，在尽量减少规则集重叠的前提下，将规则集进行分类，然后分别缓存. CRAFT 采用两级缓存结构，数据分组到来时进行逐级匹配，将匹配好的两条规则放在最后的优先级选择器进行筛选，选择优先级最高的规则进行匹配.

DevoFlow[19] 的规则分类方法是根据数据中心中流量（大象流、老鼠流）的贡献比例. DevoFlow 通过检测并提取代表这部分流的标志性字段，将需要存储在 TCAM 中的规则和需要精确匹配的规则区分开来，分开存放，进而缓解交换机 TCAM 的规则存储压力.

MMS[108] 内存交换机制，将最常匹配和最近匹配的规则存放在交换机 TCAM 中（规则的分类整理通过 EWMA[58] 等），那些最少被匹配到的规则并没有直接从 TCAM 中删除，而是被转移到控制器的 RAM 中进行缓存.

文献[109]利用动态值，将交换机空间分为两个部分，一部分区域的规则集是控制器根据交换机规则资源存储情况、历史信息和规则依赖等信息周期性离线计算下发的，属于相对静态区域；另一部分是动态缓存区域，实时根据进入交换机的数据包处理情况进行规则下发.

除了上述提到的算法外，CNOR[110]、文献[111]、BigMac[112] 也从规则分类方法、缓存优化目标等角度提出了相应的缓存机制等. 如图 20，缓存机制存在的目的是将原本交换机 TCAM 中存放的规则集区分存放，分担 TCAM 压力，因此合适的规则分类方法非常关键，另一个需要考虑的是高效的规则更新机制. 此外，缓存规则的存放位置也是机制设计要重点考虑的因素：可以在交换机内部、控制器内部，以及控制器和交换机之间设计专门的缓存层.

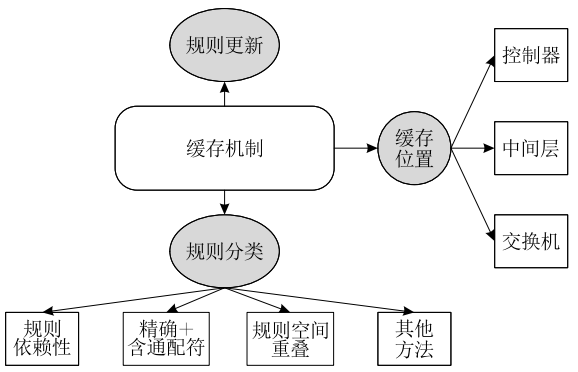

图 20　网络缓存机制研究思路

5.3　规则超时和溢出管理机制

通过控制器和交换机的协同作用，及时发现并删除交换机中冗余的超时规则，并能及时管理和控制交换机规则的溢出，避免规则溢出带来的网络性能和安全问题.

（1）规则超时处理

超时规则的处理思路一般是及时寻找交换机中未达到超时时间但又不经常被匹配使用到的规则，并进行删除或者缓存处理，防止交换机规则溢出. 如 SmartTime[113] 根据自适应的超时启发式算法，计算有效的空闲超时时间，然后主动地删除掉交换机中需要清除的规则，从而在时间和空间上更能有效利用交换机规则存储资源. FlowMaster[114] 基于马尔科夫预测模型进行学习预测，判断哪些规则更有价值，从而对规则集及时进行更新，删除不活跃的规则，释放交换机规则存储空间. STAR[115] 的规则管理模块向控制平面的路由决策模块提供网络静态统计信息，根据这些信息路由决策模块计算路由并生成转发规则下发到数据平面. 通过各个模块协同作用，STAR 能够及时清除交换机过期的规则，根据网络资源的实时分布情况，实现动态转发. IDT[116] 应用于卫星网络环境，能够智能动态分析交换机中未使用的规则，预测并删除未来较小概率使用到的规则.

（2）交换机规则溢出控制

交换机数据分组转发请求过多导致 TCAM 存储的规则和控制器更新速度不足以应对转发时，会导致 TCAM 转发规则溢出. 规则溢出会影响网络性能和安全，溢出控制机制的意义在于能够及时处理规则溢出的交换机，保障数据分组的顺利转发.

FTS[117] 可以使发生溢出的交换机继续对相应数据分组进行转发. 这个转发过程是随机的，只要与入端口不等同即可. 文章假定相邻两个交换机同时发生转发规则溢出的概率不大. 在这种情况下由于溢出交换机设置了随机转发，未匹配的数据包可以被随机转发到下一个交换机，避免了频繁与控制器交互，从而触发请求和下发规则控制信息.

控制器参与优化交换机 TCAM 规则存储资源的方案具有很高的灵活性，可以根据网络的实际情况适应性部署合适的管理模块，通过控制器自身或

者控制器与交换机的协同配合，动态地对网络转发规则的“生命周期(生成、下发、缓存、超时和溢出等过程)”进行调整和放置.其中，规则的生成和下发管理机制，根据网络的软件硬件资源特点，定制化生成合适的规则集并下发到交换机平面.网络中针对转发规则设计灵活的缓存机制是研究的热点，通过合适的规则分类分组方法，对分类后的规则集进行分布式优化放置，在保障网络正常功能的前提下，避免了所有规则直接下发存放至交换机的 TCAM 中，从而缓解交换机 TCAM 资源紧缺的问题.规则超时管理机制针对交换机中那些不经常被匹配到、仍然在交换机有效时间(timeout)的规则集.该方案的核心在于如何快速准确发现确实需要及时删除并且不影响网络性能的规则.在数据转发分组数量过多导致交换机不足以应对转发需求时，溢出管理机制可以尽可能保证交换机转发更多的数据分组.

网络转发规则管理机制大部分是通过在控制器中设计特定的模块来实现，设计灵活，扩展性强；一部分的转发规则缓存和交换机溢出控制方案等很大程度上依赖交换机的辅助实现.

6 总 结

本文从四个角度总结了如何优化 SDN 交换机中 TCAM 规则存储资源有限的问题：优化转发规则的存储结构、本地转发规则压缩算法、全局转发规则优化和控制器参与转发规则的优化管理机制.

转发规则存储结构优化的思路是在尽量保证查找性能的前提下，改进、替换 TCAM，或者多种存储介质混合存储转发规则.鉴于交换机更新周期长，涉及到硬件修改的方案在实际应用和部署的开销比较大.采用多级规则集存储可以提高 TCAM 规则存储的空间利用率，但是往往需要维护额外的跳转指令或规则，算法复杂度高，影响规则查找性能.

本地转发规则压缩一般通过软件算法减少交换机中规则占用的 TCAM 空间.压缩算法主要包括减少规则数目、裁剪规则宽度和规则的逻辑表达式最小化.该方案一个需要重点关注的工作是压缩后的规则集能否保证 TCAM 的规则匹配查找效率，以及算法是否支持快速有效地更新转发规则.

从网络全局角度缓解 TCAM 规则存储资源压力的方法包括网络端到端策略规则放置问题、路由转发策略规则压缩优化，以及两者的联合优化方案.其中，端到端策略规则的放置问题可以提炼为原始规则集的分割和相关路径上的均衡放置.对路由转发策略规则压缩一个主要方法是建立网络约束条件下的最优化方程组，然后寻找近似求解算法，或者高效的启发式算法.虽然通过标签进行路由，可以灵活、有效地避免在交换机中储存大量冗长的规则集，但是数据包携带标签字段在网络中尤其是交换机数目多、流量非常大的网络中进行转发传输，会大量消耗网络的链路带宽等资源，因此这部分研究工作的重点应放在寻找合理、有效的标签，在保证细粒度、准确转发的前提下，减少带宽的额外开销.

通过设计专门的规则产生和下发管理、网络规则缓存和超时、溢出控制模块，可以在垂直方向分布转发规则、及时释放规则存储资源以及在交换机发生规则溢出时有效保障网络性能.规则的产生和下发管理可以根据数据平面交换机异构性和规则存储资源的分布状况，适应性生成和下发规则集.网络缓存机制的设计较为灵活，研究工作也很多，主要集中在如何对原始规则集进行分类并根据网络规则存储资源进行垂直分布.转发规则的超时控制机制是通过主动或者被动预测的方式将交换机中那些不经常被匹配到的或者大概率不会被使用到的规则及时删除处理；而溢出管理机制是在交换机存储的转发规则不足以应对大量的数据转发分组时候，能尽可能保证数据分组的继续正确转发.

以上内容涵盖了 SDN 网络中，自下而上从数据平面的交换机，到控制平面的控制器，甚至整个网络范围内的设备和模块协同作用，共同缓解交换机 TCAM 规则存储资源压力，如图 21 所示.其中，转发规则存储结构优化和本地交换机压缩方法主要集中在 SDN 网络的数据转发平面，而控制器参与的转发规则管理机制则可以分别在控制平面、转发平面甚至控制和转发平面协同完成.全局网络转发规则优化方案从网络宏观角度出发，利用控制器掌握网络全局信息，实时或者离线计算网络流量路径，新路径可以在尽量保证网络性能的前提下，减少网络中交换机存储的规则总数.文章中总结的很多相关工作综合性很强，很难单纯地将文献中的方法归结到某一种特定方案中，例如 2.1.1 节提到的 E-TCAM，通过修改硬件交换机的结构，使得修改后的存储介质可以直接存储和查找含有范围字段的规则集.但它的优化目标与 3.1.2 节中提到的范围字段优化编码相似，都是为了解决范围字段在 TCAM 中直接存储时存在的范围膨胀问题.又比如 5.2 节中提到的缓存机制，核心在于有效划分规则集，使得其中一部分

的规则可以存储在缓存层中，而缓存层的存储介质可以是 TCAM、SRAM 甚至 RAM 中，这本质上也是一种 2.3 节提到的混合存储结构. 而在实际中，我们往往很难只通过一种方案就能彻底解决交换机 TCAM 规则存储资源不足的问题. 鉴于交换机设备更新周期长，在设计方案时，要尽可能避免修改交换机的硬件结构，包括转发规则的存储介质 TCAM. 软件方法的设计和创新比较灵活，易部署，但是软件算法复杂度过高时会影响交换机转发数据分组的效率. 实际环境下，我们希望充分利用 SDN 网络控制器的优势，根据全局网络拓扑和资源情况，结合需要传输的流量需求，灵活分配流量转发路径，使得尽量保证网络性能前提下减少网络规则数目. 同时结合离线或者在线的本地规则压缩算法，进一步压缩交换机中规则集的数目. 此外，规则集的缓存机制可以进一步拓展可存储规则集的空间. 最后，通过超时和溢出管理机制等处理交换机 TCAM 存储资源已满情况下的数据转发. 因此，我们需要综合考虑网络环境、网络资源分布和约束等情况，在保证网络良好的性能和服务质量前提下，从开销、效率、可行性等多角度出发，综合各类方法，提出综合性的优化策略.

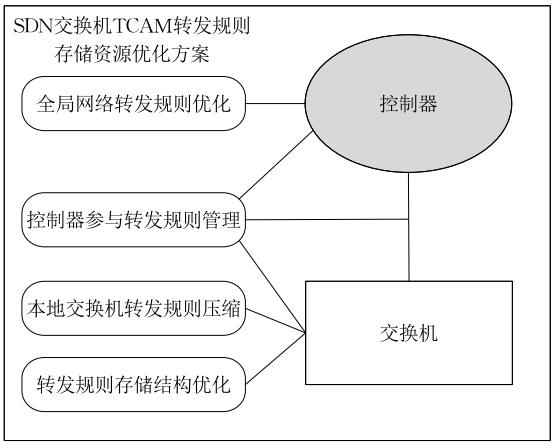

图 21　SDN 交换机 TCAM 转发规则存储资源优化

参　考　文　献

[1] Mckeown N, Anderson T, Balakrishnan H, et al. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74

[2] Li Ji-Zhou, He En. Technologies and future development trend of software-defined networking. Communications Technology, 2014, 47(2): 123-127(in Chinese)
(李纪舟，何恩. 软件定义网络技术及发展趋势综述. 通信技术，2014，47(2)：123-127)

[3] Yu F, Katz R H, Lakshman T V. Gigabit rate packet pattern-matching using TCAM//Proceedings of the 12th IEEE International Conference on Network Protocols. Berlin, Germany, 2004: 174-183

[4] Kannan K, Banerjee S. Compact TCAM: Flow entry compaction in TCAM for power aware SDN//Proceedings of the International Conference on Distributed Computing and Networking. Mumbai, India, 2013: 439-444

[5] Agrawal B, Sherwood T. Modeling TCAM power for next generation network devices//Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software. Austin, USA, 2006: 120-129

[6] Kandula S, Sengupta S, Greenberg A, et al. The nature of data center traffic: Measurements & analysis//Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement. Chicago, USA, 2009: 202-208

[7] Shalimov A, Zuikov D, Zimarina D, et al. Advanced study of SDN/OpenFlow controllers//Proceedings of the 9th Central & Eastern European Software Engineering Conference. Moscow, Russia, 2013: 1-6

[8] Srinivasan V, Varghese G, Suri S, et al. Fast and scalable layer four switching//Proceedings of the ACM SIGCOMM'98 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. Vancouver, Canada, 1998: 191-202

[9] Singh S, Baboescu F, Varghese G, et al. Packet classification using multidimensional cutting//Proceedings of the ACM SIGCOMM'03 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. Karlsruhe, Germany, 2003: 213-224

[10] Gupta P, McKeown N. Packet classification using hierarchical intelligent cuttings//Proceedings of the Hot Interconnects VII. Stanford, California, 1999: 34-41

[11] Lakshman T V, Stiliadis D. High-speed policy-based packet forwarding using efficient multi-dimensional range matching. ACM SIGCOMM Computer Communication Review, 1998, 28(4): 203-214

[12] Gupta P, Mckeown N. Packet classification on multiple fields. ACM SIGCOMM Computer Communication Review, 1999, 29(4): 147-160

[13] Srinivasan V, Suri S, Varghese G. Packet classification using tuple space search. ACM SIGCOMM Computer Communication Review, 1999, 29(4): 135-146

[14] Spitznagel E, Taylor D E, Turner J S. Packet classification using extended TCAMs//Proceedings of the 11th IEEE International Conference on Network Protocols. Atlanta, USA, 2003: 120-131

[15] Liu A X, Meiners C R, Torng E. Packet classification using binary content addressable memory. IEEE/ACM Transactions on Networking, 2016, 24(3): 1295-1307

[16] Chen T S, Lee D Y, Liu T T, et al. Dynamic reconfigurable ternary content addressable memory for OpenFlow-compliant low-power packet processing. IEEE Transactions on Circuits

and Systems I: Regular Papers, 2017, 63(10): 1661-1672

[17] Narayanan R, Kotha S, Lin G, et al. Macroflows and microflows: Enabling rapid network innovation through a split SDN data plane//IEEE European Workshop on Software Defined Networking. Darmstadt, Germany, 2012: 79-84

[18] Mogul J C, Congdon P. Hey, you darned counters! Get off my ASIC!//Proceedings of the 1st Workshop on Hot Topics in Software Defined Networks. Helsinki, Finland, 2012: 25-30

[19] Curtis A R, Mogul J C, Tourrilhes J, et al. DevoFlow: Scaling flow management for high-performance networks. ACM SIGCOMM Computer Communication Review, 2011, 41(4): 254-265

[20] Lu G, Miao R, Xiong Y, et al. Using CPU as a traffic co-processing unit in commodity switches//Proceedings of the 1st Workshop on Hot Topics in Software Defined Networks. Helsinki, Finland, 2012: 31-36

[21] Naous J, Gibb G, Bolouki S, et al. NetFPGA: Reusable router architecture for experimental research//Proceedings of the International Conference on ACM Workshop on Programmable Routers for Extensible Services of Tomorrow. Seattle, USA, 2008: 1-7

[22] Li Chun-Qiang, Dong Yong-Qiang, Wu Guo-Xin. OpenFlow table lookup scheme integrating multiple-cell Hash table with TCAM. Journal on Communications, 2016, 37(10): 128-140(in Chinese)
(李春强，董永强，吴国新. 多单元散列表与 TCAM 结合的 OpenFlow 流表查找方法. 通信学报，2016，37(10)：128-140)

[23] Tang Ya-Zhe, Zhang Yong-Qi, Yan Zi-Jian, et al. A flow table optimization scheme for software defined network. Journal of Xi'an Jiaotong University, 2018, 52(2): 13-17(in Chinese)
(唐亚哲，张永琪，颜自坚等. 面向软件定义网络的流表优化方案. 西安交通大学学报，2018，52(2)：13-17)

[24] Lee B S, Kanagavelu R, Aung K M M. An efficient flow cache algorithm with improved fairness in software-defined data center networks//Proceedings of the IEEE International Conference on Cloud Networking. San Francisco, USA, 2013: 18-24

[25] Xiong Bing, Wu Ren-Geng, Zhao Jin-Yuan, Wang Jin. Efficient differentiated storage architecture for large-scale flow tables in software-defined wide-area networks. IEEE Access, 2019, 7(1): 141193-141208

[26] Xiong Bing, Wu Ren-Geng, Zhao Jin-Yuan, Wang Jin. DAFT: A differentiated storage and accelerated lookup architecture for large-scale flow tables in OpenFlow networks. Chinese Journal of Computers, 2020, 43(3): 453-470(in Chinese)
(熊兵，邬仁庚，赵锦元，王进. DAFT：一种 OpenFlow 大规模流表区分存储与加速查找结构. 计算机学报，2020，43(3)：453-470)

[27] Cheng Y C, Wang P C. Scalable multi-match packet classification using TCAM and SRAM. IEEE Transactions on Computers, 2016, 65(7): 2257-2269

[28] Lee D, Wang C, Wu A. Bundle-updatable SRAM-based TCAM design for OpenFlow-compliant packet processor. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2019, 27(6): 1450-1454

[29] Hung C H, Wang J J, Wang L C, et al. Heterogeneous flow table integration for capacity enhancement in software-defined networks//Proceedings of the IEEE International Conference on Computing, Networking and Communications. Hawaii, USA, 2018: 832-836

[30] Ding X, Zhang Z, Jia Z, et al. Unified nvTCAM and sTCAM architecture for improving packet matching performance//Proceedings of the 18th ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems. Barcelona, Spain, 2017: 91-100

[31] Mimidis-Kentis A, Pilimon A, Soler J, et al. A novel algorithm for flow-rule placement in SDN switches//Proceedings of the 4th IEEE Conference on Network Softwarization and Workshops. Montreal, Canada, 2018: 1-9

[32] Ge J, Chen Z, Wu Y, et al. H-SOFT: A heuristic storage space optimization algorithm for flow table of OpenFlow. Concurrency and Computation: Practice and Experience, 2015, 27(13): 3497-3509

[33] Chen Z, Wu Y, Ge J, et al. A new lookup model for multiple flow tables of OpenFlow with implementation and optimization considerations//Proceedings of the IEEE International Conference on Computer and Information Technology. Xi'an, China, 2014: 528-532

[34] Liu Zhong-Jin, Li Yong, Su Li, et al. TCAM-efficient flow table mapping scheme for OpenFlow multiple-table pipelines. Journal of Tsinghua University, 2014, (4): 437-442(in Chinese)
(刘中金，李勇，苏厉等. TCAM 存储高效的 OpenFlow 多级流表映射机制. 清华大学学报(自然科学版)，2014，(4)：437-442)

[35] Pan H, Guan H, Liu J, et al. The FlowAdapter: Enable flexible multi-table processing on legacy hardware//Proceedings of the second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. Hong Kong, China, 2013: 85-90

[36] Meiners C R, Liu A X, Torng E. TCAM Razor: A systematic approach towards minimizing packet classifiers in TCAMs. IEEE/ACM Transactions on Networking, 2007, 18(2): 266-275

[37] Daly J, Liu A X, Torng E. A difference resolution approach to compressing access control lists. IEEE/ACM Transactions on Networking, 2015, 24(1): 610-623

[38] Rottenstreich O. Lossy compression of packet classifiers//Proceedings of the ACM/IEEE Symposium on Architectures for Networking and Communications Systems. Oakland, USA, 2015: 39-50

[39] Zhu H, Xu M, Li Q, et al. MDTC: An efficient approach to TCAM-based multidimensional table compression//Proceedings of the IEEE IFIP Networking Conference. Toulouse, France, 2015: 1-9

[40] Meiners C R, Liu A X, Torng E. Bit weaving: A non-prefix approach to compressing packet classifiers in TCAMs. IEEE/ACM Transactions on Networking, 2012, 20(2): 488-500

[41] Luo S, Yu H, Li L M. Fast incremental flow table aggregation in SDN//Proceedings of the IEEE International Conference on Computer Communications and Networks. Shanghai, China, 2014: 1-8

[42] Wang P, McHale L, Gratz P V, et al. GenMatcher: A generic clustering-based arbitrary matching framework. ACM Transactions on Architecture and Code Optimization, 2018, 15(4): 1-22

[43] Tsai T H, Wang K, Chao T Y. Dynamic flow aggregation in SDNs for application-aware routing//Proceedings of the IEEE International Symposium on Communication Systems, Networks and Digital Signal Processing. Prague, Czech Republic, 2016: 1-5

[44] Liu A X, Gouda M G. Complete redundancy detection in firewalls//Proceedings of the IFIP Annual Conference on Data and Applications Security and Privacy. Storrs, USA, 2005: 193-206

[45] Schiff L, Afek Y, Bremler-Barr A. Orange: Multi field OpenFlow based range classifier//Proceedings of the ACM/IEEE Symposium on Architectures for Networking and Communications Systems. Oakland, USA, 2015: 63-73

[46] Sharma S, Panigrahy R. Sorting and searching using ternary CAMs//Proceedings of the IEEE Symposium on High Performance Interconnects. Oakland, USA, 2002: 101-106

[47] Bremler-Barr A, Harchol Y, Hay D, et al. Encoding short ranges in TCAM without expansion: Efficient algorithm and applications. IEEE/ACM Transactions on Networking, 2018, 26(2): 835-850

[48] Frank G. Pulse code communication: U. S. Patent 2,632,058. 1953-3-17

[49] Demianiuk V, Kogan K. How to deal with range-based packet classifiers//Proceedings of the 2019 ACM Symposium on SDN Research. San Jose, USA, 2019: 29-35

[50] Kirill K, Sergey I N, Ori R, et al. Exploiting order independence for scalable and expressive packet classification. IEEE/ACM Transactions on Networking, 2015, 24(2): 1251-1264

[51] Anat B-B, Danny H. Space-efficient TCAM based classification using gray coding. Transactions on Computers, 2010, 61(1): 18-30

[52] Abboud A, Lahmadi A, Rusinowitch M, et al. Double mask: An efficient rule encoding for software defined networking//Proceedings of the 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops. Paris, France, 2020: 186-193

[53] Chang Y K, Hsueh C S. Range-enhanced packet classification design on FPGA. IEEE Transactions on Emerging Topics in Computing, 2015, 4(2): 214-224

[54] Karnaugh M. The map method for synthesis of combinational logic circuits. Transactions of the American Institute of Electrical Engineers Part I Communication & Electronics, 1953, 72(5): 593-599

[55] Quine W V. The problem of simplifying truth functions. The American Mathematical Monthly, 1952, 59(8): 521-531

[56] Quine W V. A way to simplify truth functions. The American Mathematical Monthly, 1955, 62(9): 627-631

[57] McCluskey E J. Minimization of Boolean functions. The Bell System Technical Journal, 1956, 35(6): 1417-1444

[58] Rudell R L. Multiple-valued logic minimization for PLA synthesis. EECS Department, University of California, Berkeley, Technical Report UCB/ERL M86/65, 1986

[59] Mcgeer R, Yalagandula P. Minimizing rulesets for TCAM implementation//Proceedings of the IEEE International Conference on Computer Communications. Rio De Janeiro, Brazil, 2009: 1314-1322

[60] Wei R, Xu Y, Chao H J. Block permutations in Boolean space to minimize TCAM for packet classification//Proceedings of the IEEE International Conference on Computer Communications. Orlando, USA, 2012: 2561-2565

[61] Wang C, Youn H Y. Entry aggregation and early match using hidden Markov model of flow table in SDN. Sensors, 2019, 19(10): 2341

[62] Braun W, Menth M. Wildcard compression of inter-domain routing tables for OpenFlow-based software-defined networking//Proceedings of the IEEE European Workshop on Software Defined Networks. Budapest, Hungary, 2014: 25-30

[63] Draves R P, et al. Constructing optimal IP routing tables//Proceedings of the IEEE INFOCOM'99. Conference on Computer Communications/18th Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320). 1999: 88-97

[64] Sun Peng-Hao, Lan Ju-Long, Lu Xiao-Yuan, et al. A field-trimming model for packet classification. Journal of Electronics and Information Technology, 2017, 39(5): 1185-1192(in Chinese)
(孙鹏浩，兰巨龙，陆肖元等. 一种基于匹配域裁剪的包分类规则集压缩方法. 电子与信息学报，2017, 39(5): 1185-1192)

[65] Jiang La-Lin, Zhang Ya-Nan, Xiong Bing. Efficient OpenFlow table splitting and compressing algorithm. Journal of Chinese Mini-Micro Computer Systems, 2018, 39(2): 310-314(in Chinese)
(姜腊林，张亚南，熊兵. 一种高效的 OpenFlow 流表拆分压缩算法. 小型微型计算机系统，2018, 39(2): 310-314)

[66] Wang Qin-Min, Zhang Zhong-Pei, Chang Qing-Mei, et al. An iterative algorithm with adjustable weight for inference channel. Journal of Electronics and Information Technology, 2012, 34(12): 2850-2854(in Chinese)

(王勤民，张忠培，常青美等. 干扰信道中一种权值可调的迭代算法. 电子与信息学报，2012，34(12)：2850-2854)

[67] Wang Xiao-Long，Liu Qin-Rang，Lin Sen-Jie，et al. Compression method based on bit extraction of independent rule sets for packet classification. Journal of Computer Applications，2018，38(336)：241-246(in Chinese)
(王孝龙，刘勤让，林森杰等. 基于独立规则集位提取的包分类压缩方法. 计算机应用，2018，38(336)：241-246)

[68] Leng B，Huang L，Wang X，et al. A mechanism for reducing flow tables in software defined network//Proceedings of the IEEE International Conference on Communications. London，UK，2015：5302-5307

[69] Chao T Y，Wang K，Wang L，et al. In-switch dynamic flow aggregation in software defined networks//Proceedings of the IEEE International Conference on Communications. Washington，USA，2017：1-6

[70] Maity I，Mondal A，Misra S，et al. Tensor-based rule-space management system in SDN. IEEE Systems Journal，2018，13(4)：3921-3928

[71] Li Xiang-Wen，Ji Meng，Cao Min，et al. An optimization scheme for resource-reuse-based OpenFlow flow table storage. Study on Optical Communications，2014，(2)：8-11(in Chinese)
(李向文，吉萌，曹敏等. 基于资源复用的 OpenFlow 流表存储优化方案. 光通信研究，2014，(2)：8-11)

[72] Kang N，Liu Z，Rexford J，et al. Optimizing the "one big switch" abstraction in software-defined networks//Proceedings of the 9th ACM Conference on Emerging Networking Experiments and Technologies. Santa Barbara，USA，2013：13-24

[73] KanizoY，Hay D，Keslassy I. Palette：Distributing tables in software-defined networks//Proceedings of the IEEE International Conference on Computer Communications. Torino，Italy，2013：545-549

[74] Sheu J P，Lin W T，Chang G Y. Efficient TCAM rules distribution algorithms in software-defined networking. IEEE Transactions on Network and Service Management，2018，15(2)：854-865

[75] Li H，Chen K，Pan T，et al. CORA：Conflict razor for policies in SDN//Proceedings of the IEEE International Conference on Computer Communications. Honolulu，USA，2018：423-431

[76] Huang H，Li P，Guo S，et al. The joint optimization of rules allocation and traffic engineering in software defined network//Proceedings of the 22nd IEEE International Symposium of Quality of Service. Hong Kong，China，2014：141-146

[77] Zhang S，Ivancic F，Lumezanu C，et al. An adaptable rule placement for software-defined networks//Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. Atlanta，USA，2014：88-99

[78] Kannan P G，Chan M C，Ma R T B，et al. Raptor：Scalable rule placement over multiple path in software defined networks//Proceedings of the IFIP Networking Conference (IFIP Networking) and Workshops. Stockholm，Sweden，2017：1-9

[79] Yoshioka K，Hirata K，Yamamoto M. Routing method with flow entry aggregation for software-defined networking//Proceedings of the IEEE International Conference on Information Networking. Da Nang，Vietnam，2017：157-162

[80] Nguyen X N，Saucez D，Barakat C，et al. OFFICER：A general optimization framework for OpenFlow rule allocation and endpoint policy enforcement//Proceedings of the IEEE International Conference on Computer Communications. Hong Kong，China，2015：478-486

[81] Gao C，Wang H，Zhai L，et al. Optimizing routing rules space through traffic engineering based on Ant colony algorithm in software defined network//Proceedings of the IEEE 28th International Conference on Tools with Artificial Intelligence. San Jose，USA，2016：106-112

[82] Ashraf U. Rule minimization for traffic evolution in software-defined networks. IEEE Communications Letters，2017，21(4)：793-796

[83] Zhang Xiao-Ning，et al. Forwarding rule multiplexing for scalable SDN-based Internet of Things. IEEE Internet of Things Journal，2018，6(2)：3373-3385

[84] Soliman M，Nandy B，Lambadaris I，et al. Source routed forwarding with software defined control，considerations and implications//Proceedings of the 2012 ACM Conference on CoNEXT Student Workshop. Nice，France，2012：43-44

[85] Soliman M，Nandy B，Lambadaris I，et al. Exploring source routed forwarding in SDN-based WANs//Proceedings of the IEEE International Conference on Communications. Sydney，Australia，2014：3070-3075

[86] Sinha Y，Bhatia S，Shekhawat V S，Chalapathi G S S. MPLS based hybridization in SDN//Proceedings of the IEEE International Conference on Software Defined Systems. Valencia，Spain，2017：156-161

[87] Guo Z，Xu Y，Cello M，et al. JumpFlow：Reducing flow table usage in software-defined networks. Computer Networks，2015，92：300-315

[88] Song H. Protocol-oblivious forwarding：Unleash the power of SDN through a future-proof forwarding plane//Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking. Hong Kong，China，2013：127-132

[89] Hu D，Li S，Huang H，et al. Flexible flow converging：A systematic case study on forwarding plane programmability of protocol-oblivious forwarding. IEEE Access，2016，4：4707-4719

[90] He C，Feng X. POMP：Protocol oblivious SDN programming with automatic multi-table pipelining//Proceedings of the IEEE Conference on Computer Communications. Honolulu，USA，2018：998-1006

[91] Hunt S，Sands D. On flow-sensitive security types. ACM SIGPLAN Notices，2006，41(1)：79-90

[92] Hu S, Chen K, Wu H, et al. Explicit path control in commodity data centers: Design and applications. IEEE/ACM Transactions on Networking, 2016, 24(5): 2768-2781

[93] MacDavid R, Birkner R, Rottenstreich O, et al. Concise encoding of flow attributes in SDN switches//Proceedings of the Symposium on SDN Research. Santa Clara, USA, 2017: 48-60

[94] Ren Y, Tsai T H, Huang J C, et al. FlowTable-free routing for data center networks: A software-defined approach//Proceedings of the IEEE Global Communications Conference. Singapore, 2017: 1-6

[95] Cohen R, Lewin-Eytan L, Naor J S, et al. On the effect of forwarding table size on SDN network utilization//Proceedings of the IEEE Conference on Computer Communications. Toronto, Canada, 2014: 1734-1742

[96] Giroire F, Moulierac J, Phan T K. Optimizing rule placement in software-defined networks for energy-aware routing//Proceedings of the IEEE Global Communications Conference. Austin, USA, 2014: 2523-2529

[97] Rifai M, Huin N, Caillouet C, et al. Too many SDN rules? Compress them with MINNIE//Proceedings of the IEEE Global Communications Conference. San Diego, USA, 2015: 1-7

[98] Zhang J, Zeng D, Gu L, et al. On rule placement for multi-path routing in software-defined networks//Proceedings of the International Conference on Collaborative Computing: Networking, Applications and Worksharing. Hangzhou, China, 2015: 59-71

[99] Wang J, Cheng S, Fu X. SDN programming for heterogeneous switches with flow table pipelining. Scientific Programming, 2018(PT. 2): 2848232. 1-2848232. 13

[100] Mu T Y, Al-Fuqaha A, Shuaib K, et al. SDN flow entry management using reinforcement learning. ACM Transactions on Autonomous and Adaptive Systems, 2018, 13(2): 1-23

[101] Huang H, Guo S, Li P, et al. Cost minimization for rule caching in software defined networking. IEEE Transactions on Parallel and Distributed Systems, 2016, 27(4): 1007-1016

[102] Geissler S, Herrnleben S, Bauer R, et al. Tablevisor 2.0: Towards full-featured, scalable and hardware-independent multi table processing//Proceedings of the IEEE Conference on Network Softwarization. Bologna, Italy, 2017: 1-8

[103] Li T, Zhou H, Luo H, et al. SAT-FLOW: Multi-strategy flow table management for software defined satellite networks. IEEE Access, 2017, 5: 14952-14965

[104] Yang G, Yu B Y, Kim S M, et al. LiteVisor: A network hypervisor to support flow aggregation and seamless network reconfiguration for VM migration in virtualized software-defined networks. IEEE Access, 2018, 6: 65945-65959

[105] Yan B, Xu Y, Xing H, et al. CAB: A reactive wildcard rule caching system for software-defined networks//Proceedings of the 3rd Workshop on Hot Topics in Software Defined Networking. Chicago, USA, 2014: 163-168

[106] Katta N, Alipourfard O, Rexford J, et al. CacheFlow: Dependency-aware rule-caching for software-defined networks//Proceedings of the Symposium on SDN Research. Santa Clara, USA, 2016: 1-12

[107] Li X, Xie W. CRAFT: A cache reduction architecture for flow tables in software-defined networks//Proceedings of the IEEE Symposium on Computers and Communications. Heraklion, Greece, 2017: 967-972

[108] Marsico A, Doriguzzi-Corin R, Siracusa D. An effective swapping mechanism to overcome the memory limitation of SDN devices//Proceedings of the IFIP/IEEE Symposium on Integrated Network and Service Management. Lisbon, Portugal, 2017: 247-254

[109] Wang D, Li Q, Jiang Y, et al. Balancer: A traffic-aware hybrid rule allocation scheme in software defined networks//Proceedings of the International Conference on Computer Communication and Networks. Vancouver, Canada, 2017: 1-9

[110] Yang C, Jiang Y, Liu Y, et al. CNOR: A non-overlapping wildcard rule caching system for software-defined networks//Proceedings of the IEEE Symposium on Computers and Communications. Natal, Brazil, 2018: 707-712

[111] Ding Z, Fan X, Yu J, et al. Update cost-aware cache replacement for wildcard rules in software-defined networking//Proceedings of the IEEE Symposium on Computers and Communications. Natal, Brazil, 2018: 00457-00463

[112] Yan B, Xu Y, Chao H J. BigMaC: Reactive network-wide policy caching for SDN policy enforcement. IEEE Journal on Selected Areas in Communications, 2018, 36(12): 2675-2687

[113] Vishnoi A, Poddar R, Mann V, Bhattacharya S. Effective switch memory management in OpenFlow networks//Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems. Mumbai, India, 2014: 177-188

[114] Kannan K, Banerjee S. FlowMaster: Early eviction of dead flow on SDN switches//Proceedings of the International Conference on Distributed Computing and Networking. Madrid, Spain, 2014: 484-498

[115] Guo Z, Liu R, Xu Y, et al. STAR: Preventing flow-table overflow in software-defined networks. Computer Networks, 2017, 125: 15-25

[116] Jan S, Guo Q, Jia M, et al. Intelligent dynamic timeout for efficient flow table management in software defined satellite network//Proceedings of the International Conference on Wireless and Satellite Systems. Harbin, China, 2019: 59-68

[117] Qiao Si-Yi, Hu Cheng-Chen, Li Hao, et al. Taming the flow table overflow in OpenFlow switch. Chinese Journal of Computers, 2018, 41(9): 2003-2015(in Chinese)
(乔思祎，胡成臣，李昊等. OpenFlow 交换机流表溢出问题的缓解机制. 计算机学报，2018, 41(9): 2003-2015)

CHEN Zhi-Peng, Ph. D. candidate. His research directions are the compression scheme and the management mechanism of forwarding rules in software defined network.

XU Ming-Wei, Ph. D., professor, Ph. D. supervisor. His research directions are network architecture, high performance routers, network security.

YANG Yuan, Ph. D., assistant professor. His major research directions are network architecture, Internet routers.

Background

Ternary Content Addressable Memory (TCAM) supports the storage of rules with wildcard and parallel lookup, which makes it become a widely used storage medium in SDN switches.

However, with the continuous expansion of network scale and network functions, the forwarding rules in the network show a growing trend. As a widely used communication standard between controllers and switches in SDN, OpenFlow protocol defines a high dimension of forwarding rules, which brings tremendous storage pressure to SDN switches using TCAM as the storage medium. This becomes a bottleneck of SDN development. In order to efficiently utilize the limited TCAM storage resources for forwarding rules, it is necessary to find optimization schemes of storage for forwarding rules in SDN. In this paper, we mainly analyze and summarize the storage optimization mechanisms of forwarding rules in SDN, from the perspectives of forwarding rule storage architecture optimization, compression of forwarding rules locally and globally, management of forwarding rules with the participation of controller(s).

Forwarding rule storage architecture optimization mainly focuses on the improvement of TCAM structure, or the integration of different storage media like TCAM, SRAM, etc. The compression of forwarding rules locally aims at decreasing the number of forwarding rules, or shortening the width of forwarding rules. However, methods for multidimensional forwarding rules compression are limited, and local compression also faces such optimization problems as rule lookup speed and update efficiency. Dynamic optimization of forwarding rules from the global view of the network balances the distribution, or decreases the number of forwarding rules among the whole network. It is flexible and efficient. Besides, it is not limited to specific network protocols such as OpenFlow. New SDN Network Architectures such as POF and P4 are also considered to further relieve TCAM rule storage pressure in switches. Management of forwarding rules with the participation of controller(s) concludes caching of forwarding rules or flow table overflow control through specifically designed modules, which are designed as a module in the controller of an additional layer between the control and data plane.

No one solution mentioned above is perfect for storing forwarding rules with TCAM in SDN. Only combine these mentioned methods rationally, and design management and optimization modules properly, can we effectively alleviate the pressure of TCAM rule storage resources in SDN.