

提高软件定义网络交换机存储能力的 流表压缩算法

沈剑良^{1,2}, 王崇越^{1,2}, 汤先拓^{1,2}, 张霞^{1,2}

(1. 中国人民解放军战略支援部队信息工程大学, 450003, 郑州;

2. 国家数字交换系统工程技术研究中心, 450002, 郑州)

摘要: 针对目前软件定义网络(SDN)中流表规模不断增大且存储效率较低的问题,提出了一种新的提高SDN交换机存储能力的流表压缩算法。首先对传统前缀编码算法分析,发现2的指数幂(2^n)范围编码后字符串最高有效位递增,且其余低位均为通配符;其次对流表中范围字段按照2的指数幂进行分割,得到多个范围字段,利用最高位和通配符实现仅用一条表项编码2的指数幂范围字段,其他范围字段采用前缀编码;最后将编码结果存储至不同三态内容寻址存储器,进行分类存储。实验结果表明,与已有的算法相比,所提流表压缩算法在独立于流表内容的基础上具备更高的更新速度与更低的最坏情况拓展比,在防火墙类型规则集编码后平均可减少86%和83%的表项数量,且不需要占用额外比特位,从更细粒度层次实现了存储空间优化使用,具有良好的应用前景。

关键词: 软件定义网络;三态内容寻址存储器;范围编码;流表

中图分类号: TP393 **文献标志码:** A

DOI: 10.7652/xjtuxb202211019 **文章编号:** 0253-987X(2022)11-0186-09

A Flow Table Compression Algorithm for Improving the Storage Capacity of Software-Defined Network Switches

SHEN Jianliang^{1,2}, WANG Chongyue^{1,2}, TANG Xiantuo^{1,2}, ZHANG Xia^{1,2}

(1. PLA Strategic Support Force Information Engineering University, Zhengzhou 450003, China;

2. National Digital Switch System Engineering and Technological Research Center, Zhengzhou 450002, China)

Abstract: This paper proposes a new flow table compression algorithm to tackle the problem of increasing flow tables in software-defined network and low storage efficiency, thus improving the storage capacity of SDN switches. Firstly, the conventional prefix coding algorithm is analyzed. The results show that the highest significant bit of the string is incremented after the exponentiation range value of 2 is encoded, and that the remaining low bits are wildcards. Secondly, the range field in the flow table is divided according to the exponential power of 2 to obtain multiple range fields. Using the highest bit and wildcards, only one table item is used to encode the exponential power range field of 2, and other range fields are prefix encoded. Finally, the coding results are stored in different ternary content addressable memories for classified storage. According to the simulation results, compared with the existing algorithms, the proposed method realizes higher update speed and lower worst-case expansion ratio on the basis of being independent of the content of the flow table. In addition, it can reduce the number of table

items by 86% and 83% on average after the firewall type rule set is coded. The range value storage based on the proposed method requires no additional bits, realizing the optimal use of storage space from a more fine-grained level, which has good application prospects.

Keywords: software-defined networking; ternary content addressable memory; range coding; flow table

随着互联网技术与信息技术革命的不断深入,旧有的网络结构无法满足灵活多变的用户需求,以软件定义网络 (software defined networking, SDN) 为首的新型网络体系具备良好的前景,受到了广泛关注。SDN 将控制平面和数据平面解耦分离,控制平面生成流表,数据平面根据流表对数据进行分类、转发等操作^[1-3],实现了网络的统一管理。利用流表的数据包分类已经被广泛应用在防火墙过滤、入侵检测、流量计费等领域^[4-5],在网络数据安全方面发挥着重要作用。

随着 SDN 及其南向协议 OpenFlow 的不断发展,流表内容愈发复杂,流表数不断增加。为了满足低时延的流式处理,三态内容寻址存储器 (TCAM) 被用于交换机中进行流表存储与匹配,但是 TCAM 存储空间较小,价格昂贵^[6],且无法直接存储具有范围值的流表项,需编码后进行存储,编码后流表项用 0, 1, * (通配符) 表示,给存储空间带来了巨大压力。

针对流表内容耗费存储空间的问题,学术界和工业界已经提出了许多方法来优化存储空间,这些研究主要集中在优化 TCAM 编码算法^[7-18]、修改 TCAM 硬件结构^[19-21]、流表优化^[22-25] 3 个方面。

TCAM 的范围编码算法是利用编码过程中流表特性来进行优化,减少编码后流表项数。最直接的前缀编码^[7] (prefix encoding, PE) 算法将每个范围值转化为前缀表示,并独立存储在 TCAM 表项中,当一条规则同时具有多个范围值时,会导致多组前缀相乘,占用大量空间,最坏情况下一条规则需要 900 个 TCAM 表项进行表示,导致范围扩展问题。通常,一个 W 位的范围最多需要 $2(W-1)$ 条前缀来表示^[7],占用大量存储空间,不满足网络的发展。为了减少编码后表项数,文献[8-13, 16]利用额外比特位来提高编码效率。文献[14-15, 17]利用规则集本身特性来提高编码效率,修改 TCAM 硬件结构则是通过修改硬件电路来提高存储空间。Yu 等^[20]提出了利用静态随机存取存储器 (SRAM) 来模拟 TCAM,可以在保留高效处理速度的前提下降低成本,提高存储空间。Gangadhar 等^[21]提出了修改 TCAM 的晶体管结构来优化 TCAM 构造,实现了

容量增加以及功耗降低,但是修改硬件电路的方法都具有较大局限性。

流表优化方法则是根据流表项之间的关系以及流表内部的信息冗余来对流表项进行压缩优化。唐亚哲等^[24]结合 TCAM 和布隆过滤器设计了混合多级流表,有效提高了存储容量。Zhong 等^[25]在几何空间上对规则集进行分析,将具有重叠特征的不同规则集进行合并转化,实现了规则集的优化,增加了存储空间利用率。

这些技术都是为了提高 TCAM 存储空间利用率,而范围编码是其中无法绕开的关键环节。针对现有范围编码算法占用额外比特位过多^[8, 11-13]或支持场景局限性过大^[9-10, 14-17]的问题,本文提出了一种不占用额外比特位的高效混合编码算法。该算法将任意 W 位范围 $[2^p, 2^q - 1]$ (其中 $p \geq 0, w \geq q > p$) 称为无范围扩展前缀编码 (none range expansion prefix encoding, NREPE) 范围,对其编码可以实现无范围扩展。对于非 NREPE 范围采用其他编码方案,由于精力限制,本文对其他范围仍采用了 PE 算法进行编码,但是在后续工作中,可以采用更高效的编码方案进行混合编码。

本文从 TCAM 的缺点着手,通过分析对比已有的优化算法,查漏补缺,首先对此前办法进行改进,设计新型编码算法,其次设计对应的架构,最后通过实验证明方法有效性。综上所述,本文提出了一种无需修改 TCAM 结构的混合编码方案,可以灵活有效地对不同类型流表进行优化;提出了一种不占用额外比特位的无范围扩展的编码算法,有效地节省了 TCAM 存储资源;从更细粒度的层次对算法编码效果进行了分析,证明了本算法高效的编码效率。

1 混合编码策略与算法实现

在本节中,对目前已有的关于 TCAM 范围编码的算法进行介绍,根据这些编码算法的独立性将其分为两类:规则集无关的范围编码和规则集相关的范围编码。

1.1 规则集无关的范围编码

最直接的独立于规则集的范围编码方式是前缀

编码算法 PE^[7],该算法直接将十进制数据转化为二进制的前缀表示形式,将所有的范围端口转化为前缀表示。算法简单易于部署,但是直接转化的范围值通常需要多条流表项表示,在最初流量规模较小时应用较多。随着网络流量不断增加,流表规模不断膨胀,PE 算法已无法满足当下网络流表存储需求。

针对 PE 算法编码效率较低的问题,文献[8]提出了栅栏编码算法,可以将任意的范围转化为一条前缀表示,但是该算法需要 $2^W - 1$ 个额外比特位来表示,在实际的部署中无法提供足够的额外比特位。进而提出的独立于数据库的范围前缀编码(database independent range preencoding, DIRPE)算法针对栅栏算法需要额外比特位过多的问题,对栅栏算法进行了改进,将范围值占用的比特位拆分成多个块,拆分后得到的各个块利用栅栏编码表示,拆分后得到的块越多,每个块包含的比特位越少,需要的额外比特位也越少,但是相应会增加得到的 TCAM 表项数。

针对 DIRPE 需要额外比特位的情况,Bremler-barr 等^[9]提出了短范围格雷码编码(short range gray encoding, SRGE)方案。SRGE 利用了二进制反射格雷码的特性来进行编码,借助二叉树的反射特性和格雷码的邻接特性,确保二叉反射树的任意范围都可以通过一个或多个折叠节点进行压缩。该算法不需要额外的比特位,可以编码不超过 1 024 位的范围,但是对于更大的范围值,则需要采用混合编码的方式处理。

Bremler-barr 等^[10]提出无扩展范围编码(range encoding with no expansion, RENE)算法,从全局出发,根据规则集中最大范围值长度对所有范围进行编码,对于较小的范围值,可以实现不需要行扩展的范围编码。该编码算法主要针对于较短的范围字段,范围字段较长时编码效率有所下降。

Demianiuk 等^[11]提出的范围减少(range reduction, RR)算法,通过将原始规则集分组后得到相互独立的不相交范围,再利用 PE 算法或 SRGE 方案进行编码后存储,在范围字段较短时编码效果较好。

文献[12]提出的分割编码结构(divide-and-conquer scheme, DCS),首次提出了分而治之的编码架构。通过对范围值进行分析,将含有范围值的规则分割为多条表项,对不同的表项采用不同的编码算法,可以有效地减少范围扩展,但是该算法需要 W 位额外比特位,在减少范围扩展的同时也占用一部分 TCAM 存储空间,在 IPV6 环境下,存储空间

优化效果并不好,仍具有较大的改进空间。

1.2 规则集相关的范围编码

规则集相关的范围编码核心在于利用规则集内部的特征来进行编码,编码效率较高,但是当规则集发生改变时,需要重新计算,代价较大,不能实现规则集的快速更新。

最基本的规则集相关的范围编码算法是比特映射算法(Bit-Map)^[16],该算法统计规则集中的所有范围规则,将不同范围用一个比特位表示,利用不同比特位对范围规则进行划分。若存在 N 个范围规则,则需要 N 个比特位来进行范围编码。该编码规则较为简单,不产生行扩展,但是缺陷较为明显,无法应用于大规模规则集。

文献[17]给出了动态范围编码算法(dynamic range encoding scheme, DRES),该算法首先对规则集中范围规则进行统计,得到每条规则的编码增益(该规则编码后可以消除的表项数)。进而按照编码增益排序,依次选择编码增益最大的规则进行编码,编码方式可以任意选择。文献[17]提供了一个系统的解决方案,虽然给出了规则集更新的方法,但是过于繁琐,更新困难。

文献[18]提出了范围特征码编码算法(RFC),该算法首先将规则集中所有范围分为多组,其次将每组范围划分为多个正交范围,最后对各组中的正交范围进行编码,通过利用额外比特位对不同范围值进行区分,从而减少了范围扩展的问题。

2 无范围扩展前缀编码

目前大多数编码算法存在占用额外比特位、编码后流表项过多等情况,针对这个问题,本文在 PE 算法和 DCS 算法的基础上提出了无范围扩展前缀编码算法,不需要占用额外比特位,对于本文提出的 NREPE 范围,可以实现无范围扩展。

2.1 无范围扩展前缀编码的基本思想

TCAM 的范围编码算法一直是研究热点,目前许多算法利用 TCAM 中的额外比特位来对范围值进行编码,但是由于空间制约,无法适应新型网络体系结构的发展。经典的 PE 算法在不需要额外比特位的基础上则会带来范围扩展问题。

本文针对 PE 算法带来的范围扩展问题,结合 PE 算法以及 DCS 算法的优点,提出了无范围扩展前缀编码,通过设计独特的编码方式可以对范围值实现不占用额外比特位的单条编码,解决了 PE 算法和 DCS 算法的缺点。通过表 1 简要描述无范围

扩展编码的思想。

表 1 范围[2,64]的 PE 编码结果
Table 1 PE encoding result of range [2,64]

原始范围	前缀编码后结果
[2,64]	000000000000001 *
[2,64]	000000000000001 **
[2,64]	000000000000001 ***
[2,64]	0000000000001 ****
[2,64]	000000000001 *****
[2,64]	0000000001000000

由表 1 知,对范围[2,64]使用前缀编码后,自上而下每条表项的最高有效位依次递进,且最高位后的比特位均使用*(通配符)代替。针对规则集编码的特性,给出了定理 1 如下。

定理 1 对于任意的 W 位范围 $[2^p, 2^q - 1]$,其中 $p \geq 0, W \geq q > p$,可以使用最高有效位递进的 $q - p$ 个向量将该 W 位范围表示。为便于表示,本文将该类型范围称为 NREPE 范围。

证明 对于任意的 W 位范围 $[2^p, 2^q - 1]$,由于 $q > p$,因此该 W 位范围必定存在区间 $[2^p, 2^{p+1} - 1]$,可以用一条前缀表示为 $0^{W-p-1}1 *^p$,此时剩余区间为 $[2^{p+1}, 2^q - 1](q > p + 1)$,仍可重复分割后编码过程,最终可使用最高有效位递进的 $q - p$ 个前缀向量将该 W 位范围表示。证明完毕。

根据定理 1 知,任意 NREPE 范围的前缀表示是有规律的,仅利用一条表项将其编码,即可解决 PE 算法范围扩展的问题,节省存储资源。

2.2 无范围扩展前缀编码流程

本文已经在 2.1 节中提出了 NREPE 范围的概念,通过分析可知,若仅利用一条表项表示 NREPE 范围,即可实现无范围扩展范围编码。规则集中的范围值,并不都是 NREPE 范围,针对该问题,本文提出一种编码架构,首先将范围值进行分割,得到 NREPE 范围和其他范围字段,对于 NREPE 范围采用无范围扩展编码,而其他类型的范围值采用前缀编码。通过混合编码技术,即可最高效率地进行编码,通过这种编码架构,可以灵活有效地支持不同的应用场景,具备良好的发展前景。

2.2.1 范围值的分割

目前网络应用向智能化、精细化发展,对数据处理有更高的要求,因此在数据包分类中端口号等范围字段也发挥着越来越重要的作用。目前数据包包头中含有的端口范围种类繁多,并不完全满足编码

要求,因此对于这些范围字段,首先需要对其进行加工处理,使其满足编码算法的要求。本文所提算法是在 PE 算法和 DCS 算法的基础上进行改进,借鉴了 DCS 算法对范围值分割的方法。即对于任意给定的范围 $[l, h](0 \leq l < h \leq 2^W - 1)$ 可以分为 3 种情况。

(1)不需要进行分割。该范围被某个 NREPE 范围覆盖($2^p < l < h \leq 2^{p+1} - 1, 0 \leq p < W$);该范围恰好为 NREPE 范围,即($l = 2^p, h = 2^q - 1, 0 \leq p < q \leq W$)。

(2)分割为两段范围。分割后均为非 NREPE 范围 $[l, 2^p - 1], [2^p, h](2^{p-1} \leq l < h \leq 2^{p+1} - 1)$;分割后为 NREPE 范围和非 NREPE 范围并存,即 $[l, 2^p - 1], [2^p, h](2^{p-1} < l, h = 2^q - 1, p + 1 < q)$ 或 $[l, 2^q - 1], [2^q, h](l = 2^p, h < 2^{q+1} - 1, p + 1 < q)$ 。

(3)分割为 3 段范围。 $[l, 2^p - 1], [2^p, 2^q - 1], [2^q, h](2^{p-1} < l \leq 2^p - 1, 2^q \leq h < 2^{q+1} - 1, p < q + 1)$,其中 $[2^p, 2^q - 1]$ 为 NREPE 范围,其余两个为非 NREPE 范围。

通过对原始范围值的分割处理,可以得到 NREPE 范围和其他范围字段,为后续打下基础。

2.2.2 规则集编码算法

对范围值进行分割后可以得到不同的范围字段,对不同的范围采用针对性编码算法,可以提高编码效率。对于 NREPE 范围,本文给出了算法 1 来进行编码,而对于其他范围字段,本文采用前缀编码来进行表示。

算法 1 NREPE 范围编码算法

输入 范围值的最高位与最低位 p, q 以及位宽 W
($p + 1 < q, p, q$ 构建了一个 NREPE 范围 $[2^p, 2^q - 1]$)

输出 字符串向量 s

- 1. 初始化一个空的 W 位字符串向量 s
- 2. 利用 q 设置字符串向量的最高位
- 3. 从字符串最高位 $W - 1$ 到 $q - 1$ 进行遍历填充,对第 $q - 1$ 位填充为 *, 其余高位填充为 0
- 4. for all i from $W - 1$ to $q - 1$ do
- 5. if $i = q - 1$ then
- 6. $s.append(' *')$ { ' * ' 为通配符 }
- 7. else
- 8. $s.append('0')$;
- 9. $i--$;
- 10. 利用 p 设置字符串向量的最低位
- 11. 从字符串 $q - 2$ 位到 0 进行遍历填充,对第 p 位

填充为 1,其余位填充为 *

```

12. for all  $i$  from  $q-2$  to 0 do
13. if  $i=p$  then
14.      $s.append('1')$ 
15. else
16.      $s.append('*')$ ;
17.      $i--$ ;
18. return  $s$ ;

```

通过对规则集中的范围值进行分割编码后,可以实现不占用额外比特位的目的。由于对范围值采用了独特的编码方式,为了实现流表项的匹配,同样需要对包头向量进行对应的搜索密钥编码,下面给出搜索密钥算法的伪代码。

算法 2 NREPE 搜索密钥编码算法

输入 端口值 v 以及位宽 W

输出 字符串向量 k

```

1. 初始化一个空的  $W$  位字符串向量  $k$ 
2. 将端口值使用二进制表示,最高有效位设为  $y$ 
3. 从  $k$  的最高位开始遍历填充, $y$  左侧填充为 0, $y$ 
   位填充为 1,其余填充为 *
4. for all  $i$  from  $W-1$  to 0 do
5. if  $i=y$  then
6.      $k.append('1')$ 
7.   for all  $j$  from  $y-1$  to 0 do
8.        $k.append('*')$ 
9.        $j--$ 
10.    break
11. else
12.      $k.append('0')$ 
13.      $i--$ 
14. return  $k$ ;

```

通过设计范围编码算法以及密钥算法,可以有效地减少范围扩展问题。本文所提出算法的核心在于利用 p 和 q 对范围规则进行表示,使用 p 表示范围下限, q 表示范围上限。在搜索密钥编码算法中,则利用端口值的最高有效位进行编码,可以实现精确匹配,本文在图 1 中使用了范围值 $[5, 71]$ 作为例子,完整地展示了如何进行范围分割,范围编码以及搜索密钥编码。

在图 1(a)中,对范围值进行了分割,将其分为 3 个范围字段,其中 $[8, 63]$ 是 NREPE 范围。在图 1(b)中,针对其中的 NREPE 范围进行了编码,在图 1(c)中,分别使用 15 和 7 进行了测试,结果显示 15 被编码后符合规则,7 因最高有效位不匹配而不符

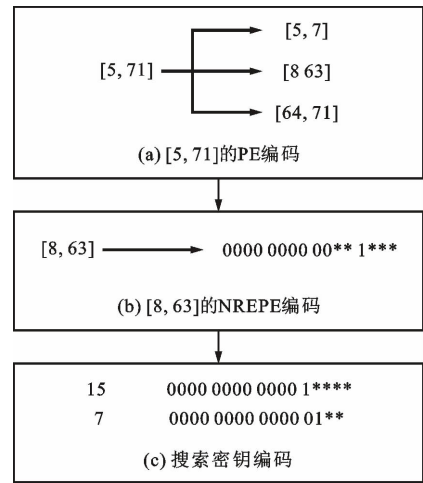


图 1 NREPE 范围编码和搜索密钥编码示例

Fig. 1 NREPE range coding and search key coding examples

合规则。

2.3 NREPE 的结构体系

在 2.2 节中已经介绍了 NREPE 编码的基本流程,即通过对范围值分割后进行针对性编码来减少范围扩展,不同的类型范围值采用不同的编码方式。为了更好地部署本文提出的算法并保留未来支持新型算法的能力,本文提出了一个开放和灵活的架构,可以使 TCAM 获得更高效的存储效率。该框架无需对交换机硬件进行改造,通过软件方法对规则进行压缩。

图 2 展示了 NREPE 编码在数据平面的结构,当数据包进入后,首先被数据包处理器所解析,提取特定字段组合后形成包头字段;其次将包头字段输入编码模块,通过搜索关键字编码算法进行编码,进而将编码结果传输至匹配模块(TCAM 块)匹配,最终输出匹配结果。当规则集或编码方式改变时,可以通过控制平面下达修改指令,并对 TCAM 块中的流表项进行重配置,实现动态更新。由于本算法仅针对范围值本身,对于未来可能出现的其他字段的范围值,该编码算法仍然有效,因此所提出的结构仍保留广阔的发展空间,适应未来网络发展。

由于 TCAM 匹配的目的是对输入的数据包进行转发、丢弃、分类等处理。与 TCAM 进行匹配的

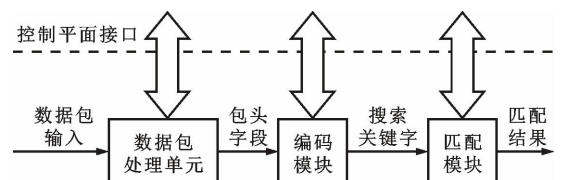


图 2 NREPE 结构框架

Fig. 2 NREPE structural framework

是从数据包中提取的多元组,编码算法也是仅针对多元组,并不影响数据包本身结构,因此并不存在解码这一环节。

首先,对规则集进行处理,将其中的范围分割后形成非 NREPE 范围规则和非 NREPE 范围规则,对于 F 个任意的范围值 $R_i(1 \leq i \leq F)$,需要将其分割为 j 个子范围,用 $R_i^j(1 \leq j \leq 3)$ 表示。最终得到 $\sum_{i=1}^F \sum_{j=1}^J R_i^j$ 个子范围,并对其进行编码。然后将整个 TCAM 划分为多个块,将编码后得到的规则集存储至 TCAM 块中,由于采用了混合编码方式,采用不同编码方式的 TCAM 块也需要组合。

由于本文将范围值分割为 NREPE 范围和非 NREPE 范围,二者编码方式不同,因此存储后的 TCAM 分为两种类型,分别存储 NREPE 和 PE 编码的范围值,两两组合后共存在 4 种不同类型的

TCAM 块。

3 实验评价

在本节中测试了本文提出算法的性能,由于真实的分类数据集数据难以得到,本文数据是通过 ClassBench^[26]生成的模拟真实分类数据集。ClassBench 是由华盛顿大学开发的实验平台,它是一套对数据包分类算法和设备进行测试的工具,可以生成精确模拟真实数据集的实验规则库,主要有 ACL(访问控制列表)、FW(防火墙规则)、IPC(IP 链)3 种规则。本文通过 ClassBench 生成了 10 组 300 000 规模的规则集,并在个人 PC 上利用 Matlab 对数据进行了模拟仿真,由于生成的规则集存在冗余情况,因此消除冗余后部分规则集数小于 300 000,但是并不影响后续的测试。对生成的数据集进行统计后,结果如表 2 所示。

表 2 ClassBench 产生的规则集
Table 2 Rule sets of ClassBench

规则集	源端口范围 规则数	目的端口范围 规则数	源和目的端口 范围规则数	源和目的端口 非范围规则数	不同类型 规则数
FW1	4 930	4 653	19 469	230 669	2
FW2	56 447	0	0	238 851	1
FW3	2 917	1 796	14 208	205 709	2
FW4	129 700	28 588	26 199	91 243	9
FW5	3 394	1 985	9 801	224 090	3
ACL1	0	36 002	0	263 220	12
ACL2	0	8 650	0	113 450	2
ACL3	0	80 324	0	219 103	16
ACL4	0	75 209	0	224 058	19
IPC1	7 603	29 404	0	262 190	4

对规则集统计后得到,FW 类型的规则集在源和目的端口中均含有范围类型,然而其范围种类较少。ACL 类型的规则集仅在目的端口具有范围,这些范围字段种类相较于 FW 和 IPC 更多。IPC 类型的规则集虽然在源和目的端口都具有范围字段,但源和目的端口并不同时具备。

3.1 拓展比性能比较

为了更好地展示本文提出方法的优势,本文选择了 PE 算法、DIRPE 算法和 DCS 进行比较。比较时使用占用额外比特位数、更新速度、最坏情况扩展比 ER(expansion ratio)作为依据,扩展比 ER 定义为具有范围的规则经过范围编码后所得的规则数 n_c 比编码前的规则数 n_d 。

由于本算法不改变 TCAM 结构以及芯片设计,仅利用软件对规则集以及提取的包头向量进行编码,因此所造成的额外功耗相比于 TCAM 本身功耗几乎可以忽略,而算法在对原始规则集进行编码并存储后重点在于规则集发生改变时的更新速度,对于更新速度以及其他指标在表 3 中进行了对比。

为了能够清楚地表示本文所提出方法针对不同规则集源和目的端口范围的编码能力,在本节中分别针对源端口和目的端口进行了编码。图 3 展示了仅在源端口进行编码的结果,从图 3 中可以看出,当仅对源端口的范围进行编码的时候,所提出的 NREPE 算法相较于 PE、DIRPE 以及 RENE 算法可以有效地降低扩展比,在 FW 和 IPC 类型的规则

集中扩展比基本都为1,这是因为在FW和IPC规则集中包含了大量[1 024, 65 535]范围值,而NREPE对此种类型的数据可以进行无范围扩展编码。ACL规则集中并不包含源端口具有范围的规则,因此并未在图3中出现。

表3 各编码算法比较

Table 3 Comparison of coding algorithms

编码算法	额外位/b	最坏情况拓展比	增量更新成本
PE	0	$2W-2$	$O(W^F)$
DIRPE (k 个比特块)	36	$2W/k-1$	$O(W/k^F)$
DCS	16	1	$O(3^F)$
RENE	16	$W\lg(W)+1$	$O(3^F)$
NREPE	0	1	$O(3^F)$

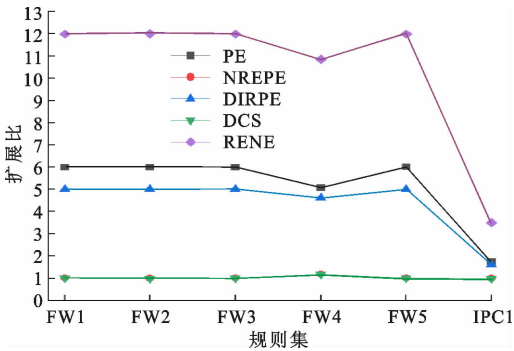


图3 对源端口进行编码的扩展比
Fig. 3 Expansion ratio on source port

图4展示了仅在目的端口进行范围编码的结果。可以看出,NREPE算法在各种规则集下都优于传统的PE算法,并且都有较大幅度的领先,但在ACL2数据集下优化效果并不十分突出,这是因为在该规则集中包含大量[1 025, 65 535]范围值,对于[1 025, 65 535]范围值,需要将其分割为[1 025, 2 047]、[2 048, 65 535]两种情况,其中[2 048,

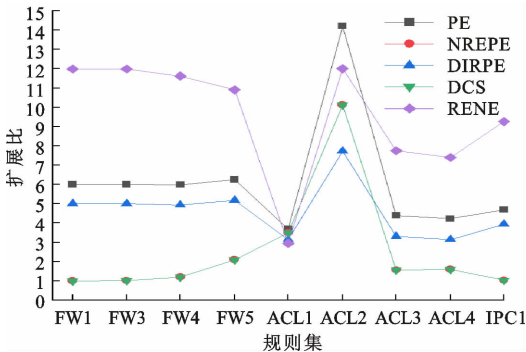


图4 对目的端口进行编码的扩展比
Fig. 4 Expansion ratio on destination port

65 535]仅需要一条表项表示,而[1 025, 2 047]需要多条表项表示,因此最终仍需要多条表项,但相较于PE算法依然提升了40%的空间利用率。RENE算法表现不佳,因为其适用于较短范围的范围编码,当存在大量较长范围时表现较差。

图5给出了同时对源和目的端口进行范围编码的结果。在两个端口均为范围的情况下,需要根据范围是否可以分割为NREPE范围进行分块存储。对于只有一个端口为范围的规则集,编码后所需的TCAM表项和针对该端口进行范围编码所得到的结果一致。例如在ACL规则集中,源端口均不包含范围,因此同时对源端口和目的端口编码的结果与针对目的端口编码得到的结果一致,但在FW规则集中,同时对源和目的端口编码后结果会发生巨大变化。在FW类型的规则集中,使用NREPE算法进行编码所得到的扩展比较PE算法平均降低了86%,相较于DIRPE算法降低了83%,效果十分明显。

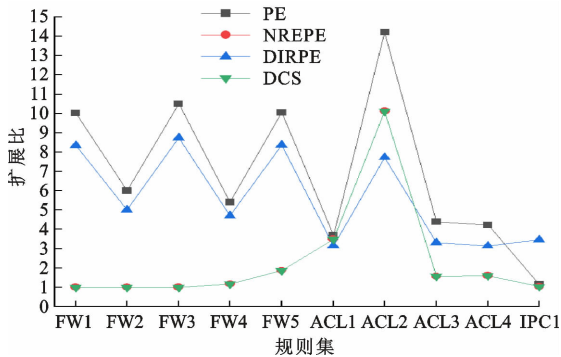


图5 对源和目的端口均进行编码的扩展比
Fig. 5 Expansion ratio on either port

这是由于FW中包含了大量的[1 024, 65 535]范围值,扩展比近似为1,并且源和目的端口均包含范围规则。如果采用其他方法,会导致大量的范围扩展,在ACL规则集中,NREPE编码后所需的TCAM表项相较PE和DIRPE算法平均降低了37%和4%,这是因为在ACL1数据集中包含了大量的范围长度较小的规则,3种算法编码效果较为相似,而在ACL2中3种算法效果都不十分理想。在IPC数据集中,NREPE算法相较于PE和DIRPE可以分别降低11%和71%的表项数。由于RENE算法在FW类型中表现不佳,通过实验知其源和目的端口同时编码时扩展比远超其他类型算法,扩展比基本均在100以上,因此并未在图中展示。

3.2 比特位数性能比较

NREPE算法相较于其他算法可以有效地降低

扩展比 ER,但相比于 DCS 算法,仅分析扩展比并无明显优势。本文所提出算法并不需要额外比特位,而规则集在 TCAM 中的存储在更细粒度上是对比特位进行存储,仅利用 ER 并不能突出本算法的优势,因此本文提出了比特位数比(bit number ratio, BNR)作为补充比较指标,从更细粒度的层次评价算法的编码效果。BNR 定义为编码后所需要的比特位数 $n_{e,b}$ 比编码前占用的比特位数 $n_{d,b}$ 。由于篇幅限制,本文仅给出了对同时从源和目的端口进行编码后的比特位数分析,结果如图 6 所示。

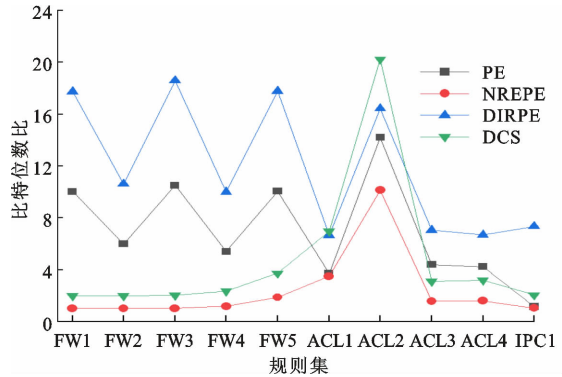


图 6 源和目的端口均进行编码的比特位数比
Fig. 6 Bit number ratio on either port

根据实验结果可知,本文所提出的 NREPE 编码算法占用的比特位数远小于其他算法,即使相比效果最好的 DCS 算法,平均占据的比特位也减少了 50%。通过综合比较,本文所提出的算法对存储空间优化效果最好。

4 结束语

本文提出了一种提高软件定义网络交换机存储能力的流表压缩算法。该算法针对 SDN 交换机的流表存储模块,对流表项进行分割后混合编码,实现了不占用额外比特位的低开销无范围扩展编码,有效提高了存储能力,且是一种通用算法,可以针对任意类型范围值,适用性广。从更细粒度利用存储资源,易于部署。实验结果表明,本文提出的流表压缩算法在各种规则集下,均可有效地优化存储空间,具备良好的扩展性,且是一种完整的编码架构,能够有效地适应网络流量的变化与发展。

参考文献:

[1] ABBASI M, SHOKROLLAHI A, KHOSRAVI M R, et al. High-performance flow classification using hybrid clusters in software defined mobile edge compu-

ting [J]. Computer Communications, 2020, 160: 643-660.

[2] CHAUDHARY R, AUJLA G S, KUMAR N, et al. Optimized big data management across multi-cloud data centers: software-defined-network-based analysis [J]. IEEE Communications Magazine, 2018, 56(2): 118-126.

[3] 杨洋, 吕光宏, 赵会, 等. 深度学习在软件定义网络研究中的应用综述 [J]. 软件学报, 2020, 31(7): 2184-2204.

YANG Yang, LÜ Guanghong, ZHAO Hui, et al. Survey on deep learning applications in software defined networking research [J]. Journal of Software, 2020, 31(7): 2184-2204.

[4] 周亚东, 陈凯悦, 冷俊园, 等. 软件定义网络流表溢出脆弱性分析及防御方法 [J]. 西安交通大学学报, 2017, 51(10): 53-58.

ZHOU Yadong, CHEN Kaiyue, LENG Junyuan, et al. Analysis and a defense method for overflow vulnerability of flow tables in software defined networks [J]. Journal of Xi'an Jiaotong University, 2017, 51(10): 53-58.

[5] ZHENG Ling, JIANG Jing, PAN Weitao, et al. High-performance and range-supported packet classification algorithm for network security systems in SDN [C]//2020 IEEE International Conference on Communications Workshops (ICC Workshops). Piscataway, NJ, USA: IEEE, 2020: 1-6.

[6] SRINIVASAVARMA V S M, VIDHYUT S, MAHAMMAD S. A TCAM-based caching architecture framework for packet classification [J]. ACM Transactions on Embedded Computing Systems, 2021, 20(1): 2.

[7] SRINIVASAN V, VARGHESE G, SURI S, et al. Fast and scalable layer four switching [J]. ACM SIGCOMM Computer Communication Review, 1998, 28(4): 191-202.

[8] LAKSHMINARAYANAN K, RANGARAJAN A, VENKATACHARY S. Algorithms for advanced packet classification with ternary CAMs [C]// Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. New York, NY, USA: ACM, 2005: 193-204.

[9] BREMLER-BARR A, HENDLER D. Space-efficient TCAM-based classification using gray coding [J]. IEEE Transactions on Computers, 2012, 61(1): 18-30.

- [10] BREMLER-BARR A, HARCHOL Y, HAY D, et al. Encoding short ranges in TCAM without expansion: efficient algorithm and applications [J]. IEEE/ACM Transactions on Networking, 2018, 26(2): 835-850.
- [11] DEMIANIUK V, KOGAN K. How to deal with range-based packet classifiers [C]//Proceedings of the 2019 ACM Symposium on SDN Research. New York, NY, USA: ACM, 2019: 29-35.
- [12] SUN Hai, SUN Yan, VALGENTI V C, et al. TCAM-based classification using divide-and-conquer for range expansion [C] // 2014 23rd International Conference on Computer Communication and Networks (ICCCN). Piscataway, NJ, USA: IEEE, 2014: 1-8.
- [13] LI Wenjun, LIU Xinwei, LE Wenxia, et al. A practical range encoding scheme for TCAMs [C]//Proceedings of the ACM SIGCOMM 2019 Conference Posters and Demos. New York, NY, USA: ACM, 2019: 163-165.
- [14] ABBASI M, VAKILIAN S, FANIAN A, et al. Ingredients to enhance the performance of two-stage TCAM-based packet classifiers in internet of things: greedy layering, bit auctioning and range encoding [J]. EURASIP Journal on Wireless Communications and Networking, 2019, 2019(1): 287.
- [15] 田乐, 陈庶樵, 黄慧群, 等. 利用域转换的三态内容寻址存储器报文分类算法 [J]. 西安交通大学学报, 2013, 47(10): 97-102.
TIAN Le, CHEN Shuqiao, HUANG Huiqun, et al. A packet classification approach for ternary content-addressable memory using domain transformation [J]. Journal of Xi'an Jiaotong University, 2013, 47(10): 97-102.
- [16] LIU Huan. Efficient mapping of range classifier into ternary-CAM [C]//Proceedings of the 10th Symposium on High Performance Interconnects. Piscataway, NJ, USA: IEEE, 2002: 95-100.
- [17] CHE Hao, WANG Zhijun, ZHENG Kai, et al. DRES: dynamic range encoding scheme for TCAM coprocessors [J]. IEEE Transactions on Computers, 2008, 57(7): 902-915.
- [18] SUN Penghao, LAN Julong, WANG Peng, et al. RFC: range feature code for TCAM-based packet classification [J]. Computer Networks, 2017, 118: 54-61.
- [19] NORIGE E, LIU A X, TORNG E. A ternary unification framework for optimizing TCAM-based packet classification systems [J]. IEEE/ACM Transactions on Networking, 2018, 26(2): 657-670.
- [20] YU Weiwen, SIVAKUMAR S, PAO D. Pseudo-TCAM: SRAM-based architecture for packet classification in one memory access [J]. IEEE Networking Letters, 2019, 1(2): 89-92.
- [21] GANGADHAR A, BABULU K. Design of low-power and high-speed CNTFET-based TCAM cell for future generation networks [J]. The Journal of Supercomputing, 2021, 77(9): 10012-10022.
- [22] ROTTENSTREICH O, TAPOLCAI J. Optimal rule caching and lossy compression for longest prefix matching [J]. IEEE/ACM Transactions on Networking, 2017, 25(2): 864-878.
- [23] 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 [J]. 计算机应用, 2018, 38(8): 2375-2380.
WANG Xiaolong, LIU Qinrang, LIN Senjie, et al. Compression method based on bit extraction of independent rule sets for packet classification [J]. Journal of Computer Applications, 2018, 38(8): 2375-2380.
- [24] 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 [J]. 西安交通大学学报, 2018, 52(2): 13-17.
TANG Yazhe, ZHANG Yongqi, YAN Zijian, et al. A flow table optimization scheme for software defined network [J]. Journal of Xi'an Jiaotong University, 2018, 52(2): 13-17.
- [25] ZHONG Jincheng, CHEN Shuhui. Efficient multi-category packet classification using TCAM [J]. Computer Communications, 2021, 169: 1-10.
- [26] TAYLOR D E, TURNER J S. ClassBench: a packet classification benchmark [J]. IEEE/ACM Transactions on Networking, 2007, 15(3): 499-511.

(编辑 武红江 刘杨)