doi:10.11959/j.issn.1000-436x.2017048

基于信息熵的匹配域裁剪算法

孙鹏浩，兰巨龙，张少军，李军飞
（国家数字交换系统工程技术研究中心，河南 郑州 450002）

摘 要：随着网络功能日益多样化，分组分类技术对匹配域数量、表项深度等需求不断提高，加剧了硬件存储压力。为保证查表效率和硬件资源利用率，提出基于信息熵的匹配域裁剪算法。通过分析匹配域冗余信息，提出匹配域裁剪模型；通过分组头部信息熵的映射建模，将匹配域裁剪算法复杂度从 NP 难降为线性复杂度。实验结果表明，较现有方案，所提方案所需三态内容寻址存储器（TCAM，ternary content-addressable memory）存储空间能够进一步减少 40%以上，或随着流表规模增长，所提案能够明显减少算法运行时间。
关键词：分组分类；三态内容寻址存储器；OpenFlow；信息熵
中图分类号：TP393 **文献标识码**：A

Information entropy based match field cutting algorithm

SUN Peng-hao, LAN Ju-long, ZHANG Shao-jun, LI Jun-fei
(National Digital Switching System Engineering and Technological R&D Center (NDSC), Zhengzhou 450002, China)

Abstract: With the increasing diversity of network functions, packet classification had a higher demand on the number of match fields and depth of match table, which placed a severe burden on the storage capacity of hardware. To ensure the efficiency of matching process while at the same time improve the usage of storage devices, an information entropy based cutting algorithm on match fields was proposed. By the analysis on the redundancy of match fields and distribution pattern in a rule set, a match field cutting model was proposed. With the mapping of matching process to the process of entropy reduction, the complexity of optimal match field cutting was reduced from NP-hard to linear complexity. Experiment results show that compared to existing schemes, this scheme can need 40% less TCAM storage space, and on the other side, with the growing of table size, the time complexity of this algorithm is also far less than other algorithms.
Key words: packet classification, TCAM, OpenFlow, information entropy

1 引言

分组分类技术在防火墙、访问控制等多种网络应用中扮演着重要的角色，也是下一代互联网设备和新型网络服务实现的关键技术之一。分组分类技术的适用规模和速度直接影响到网络技术的性能。为保证查表速度和灵活性，基于 TCAM 芯片的查表方法成为实现分组分类技术的主流技术，因而研究高效的基于 TCAM 分组分类规则存储算法，对于提升网络应用性能、扩大网络应用规模具有重要意义[1]。

由于 TCAM 芯片的资源开销高、功耗大[2]等缺点，目前在大规模网络中 TCAM 的应用受到了一定的限制。而在存储内容上缺乏有效的优化，又使 TCAM 的存储压力进一步增加：随着以 OpenFlow[3] 为代表的一系列新型网络协议的出现，分组分类规则集的规模不断增大，匹配域数量不断增多，增加了表项宽度需求，如 OpenFlow v1.3 协议共

收稿日期：2016-10-25；**修回日期**：2017-01-17
基金项目：国家重点基础研究发展计划（“973”计划）基金资助项目（No.2013CB329104）；国家自然科学基金资助项目（No.61521003）；国家高技术研究发展计划（“863”计划）基金资助项目（No.2015AA016102）
Foundation Items: The National Basic Research Program of China (973 Program) (No.2012CB315901), The National Natural Science Foundation of China (No.61372121), The National High Technology Research and Development Program of China (863 Program) (No.2015AA016102)

支持 40 个匹配域，总宽度为 1 127 bit，远远超过传统分组分类规则宽度；对于 TCP、UDP 等协议常用的端口号范围匹配模式，一条规则可能占用多条 TCAM 表项，引起了范围扩张（range expansion）问题[4]，这对 TCAM 中的表项存储深度提出了更高的要求。

目前，针对以上问题，基于 TCAM 的表项存储压缩技术存在着广泛的研究基础。对于范围匹配引起的范围扩张问题，主要的解决方案集中于通过改变编码方式来提高 TCAM 利用率。文献[4]提出了“栅栏编码”（fence code）方式，大量使用 TCAM 中每行的多余比特进行编码，但占用的表项宽度过大；文献[5]提出了基于二维对称格雷码编码方式，通过提升相邻编码的近似性来提升不同编码的合并率，对于多余比特仍有较大消耗；文献[6]重点着眼于提高多余比特利用率，提出了高效的多余比特编码方式；文献[7]在二维对称格雷码编码的基础上进行了改进，通过增加一定程度的多余比特与格雷码进行混合编码，对于短范围的扩张问题起到了较好的控制作用。在不同表项之间的合并压缩方面，Meiners[8]提出了新的编码特征提取算法，能够提取不同表项的相似性并且进行合并，从而在一定程度上减小总体表项深度；文献[9]通过多维空间的方式构建表项视图，将表项进行分割和再合并，减少了表项的数量。PC-DUOS[10]、PC-DUOS+[11]和 PC-TRIO[12]通过改进单级 TCAM 匹配的方式，对协议树的分级提取不同索引值，从而实现多级 TCAM 的索引值匹配模式，以大规模协议树分析的计算量为代价减少了表项规模。H-SOFT[13]将多匹配域规则集进行了拆分，按照不同匹配模式分组进行匹配操作，算法较为复杂，且可扩展性较弱。文献[14,15]基于对规则集的特征分析，提出了匹配域提取方案，然而算法复杂度太高，难以应用于 OpenFlow 等多匹配域协议中。

基于以上问题，为实现高效的 TCAM 存储空间压缩算法，本文通过引入信息熵的概念，对不同匹配域之间冗余关系和相关度进行分析建模，提出了基于信息熵的匹配域裁剪（EC, information entropy based match field cutting）方案，相比于现有方案，在压缩效率和算法运行效率上都具有明显的提高。

2　匹配域冗余分析

在网络设备数据平面，分组分类规则集通常以 $<F,A>$ 的格式存储。其中，F 表示匹配域，由不同的分组头部字段按照一定的顺序组成，如 IP 源地址、IP 目的地址、TTL 等；A 表示动作指令，存放相关数据分组的处理方式，如转发、丢弃、上报控制器等。数据分组在网络设备中的匹配过程在于提取数据分组头部信息，生成与 F 格式相同的分组头部域，与规则集中的规则的匹配域部分进行逐一对比，以相似度最高的规则作为匹配结果，从而执行此规则对应的执行动作 A。本文研究内容主要着眼于匹配域 F 的规则集合，现将本文中主要使用的变量定义如下：对于一个由 N 条规则组成的规则集 χ，每条规则记作 $R_1,R_2,\cdots,R_N$，其中，对于全部匹配域所包含的 m 个字段，记作 $F_1,F_2,\cdots,F_m$，若一个数据分组头部 P 与规则 R 匹配，则记作 $P\rightarrow R$。

通常，若 $\exists\ R_i,R_j\in\chi(i\neq j)$，$R_i$、$R_j$ 之间没有重叠，则称 R_i 与 R_j 正交，记作 $R_i\perp R_j$。对于采用前缀匹配的规则，如 IP 地址的最长前缀匹配原则，同等长度的前缀地址之间必然正交（否则为相同规则）；对于许多规则集（如 OpenFlow），为了便于控制层的管理，通常规则的正交性也是默认的流表下发准则；而对于常见的访问控制列表，TCP/UDP 端口号范围则经常出现重叠。规则的正交性会影响 TCAM 中的匹配输出结果（常见 TCAM 在同等匹配程度下，优先输出低地址的表项索引），因此对于非正交规则之间重叠的部分，必须保证原规则的存放顺序，否则数据平面会对数据分组做出错误处理。

定义 1　对于规则集 G，若 $\forall R_i,R_j\in G(i\neq j)$，交换 R_i、R_j 在 G 中的存储顺序不改变 G 对于所有数据分组头部的匹配结果，则称 G 满足顺序无关性，或者称 G 是一个无序集；反之，则称 G 是一个有序集。

定理 1　若一个规则集 G 是无序集，则其中任意规则之间没有重叠；反之，若一个规则集中的任意规则之间没有重叠，则这个规则集是无序集。

证明　1) 充分性。对于一个无序集 G，若其中存在规则重叠，即 $\exists R_i,R_j\ (R_i,R_j\in G)$，$R_i\ \&\ R_j$ 不为空，则重叠部分规则记作 R'，即 $R_i\ \&\ R_j=R'$。例如，R_i 表示端口号[0, 1 024]，R_j 表示端口号[137，139]，则 R' 为[137,139]。对于数据分组头部 $P\rightarrow R'$，若 R_i、R_j 按照 R_i 优先于 R_j 的顺序存放，则规则集 G 对于数据分组头部 P 的匹配结果为 R_i，若按照 R_j 优先于 R_i 的顺序存放，则规则集 G 对于数据分

组头部 P 的匹配结果为 R_j，则 G 不是无序集，与假设矛盾。

2) 必要性。对于一个规则集 G，若 G 不是无序集，则其中至少存在一组 R_i、R_j 和一个数据分组头部 P，使当 G 中的存放顺序 R_i 优先于 R_j 时，P 的匹配结果为 R_i，反之为 R_j。因此，有 $P \to R_i$ 且 $P \to R_j$，即 $P \to R_i \,\&\, R_j$，R_i、R_j 之间存在重叠部分。所以，若 $\forall R_i, R_j \in G(i \neq j)$，有 $R_i \perp R_j$，则规则集 G 为无序集。

匹配域的裁剪需在无序集上进行，以避免重叠规则在裁剪后有难以处理的冲突问题。图 1 所示的规则集包含 $R_1 \sim R_5$ 这 5 条规则，每条规则有 $F_1 \sim F_3$ 个匹配域，其中，F_2 和 F_3 为范围匹配。从图 1 可以看出，R_1 与 R_5 存在规则重叠，即对于数据分组头部取值 $<F_1, F_2, F_3> = <1,[2,4],[3,5]>$ 时，R_1 与 R_5 都与此数据分组匹配，因此，此规则集为有序集。而通过将 R_5 分离，将原规则集拆分为由 $R_1 \sim R_4$ 组成的子集和由 R_5 组成的子集，则此时 2 个规则子集都成为无序集。

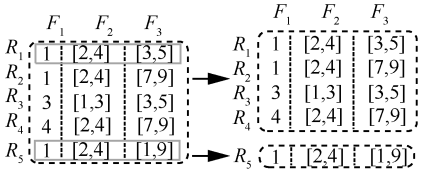


图 1 规则集分组查询示意

图 2 所示规则集为图 1 中 $R_1 \sim R_4$ 组成的无序集。存在 3 个匹配域 $F_1 \sim F_3$，其中，F_1 能够区分规则 R_3、R_4，以及 R_1、R_2 组成的子集；F_2 能够区分 R_3 或者 R_1、R_2、R_4 组成的子集；F_3 能够区分 R_1、R_3 组成的子集和 R_2、R_4 组成的子集。显然，对单独一个匹配域来说，$F_1 \sim F_3$ 这 3 个匹配域中 F_1 对整个规则集中规则的区分度最大。而对于 F_1 所不能区分的规则 R_1 和 R_2，F_3 能够进行区分而 F_2 不能。因此，取 F_1 和 F_3 这 3 个匹配域就能完成对整个规则集的区分，即 F_2 可以被“裁剪掉”而不会带来新的规则冲突。

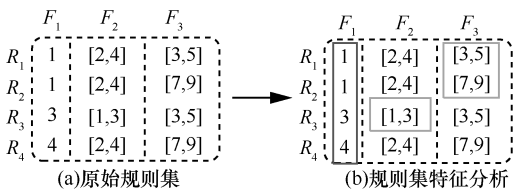


图 2 匹配域冗余示意

定理 2 对于包含 m 个匹配域的规则集 χ，提取出其中 $m-k$（$k<m$）个匹配域组成新的规则集 χ^{m-k}（规则排列顺序不变），若数据分组头部 $P \to R_i$（$R_i \in \chi$），则 $P \to R_i^{m-k}$（$R_i^{m-k} \in \chi^{m-k}$）。

证明 若 $P \to R_i$，则 P 中各字段都匹配于对应 R_i 中的各个匹配域取值 $f_1, f_2, \cdots, f_m$，而 R_i^{m-k} 中的所有匹配域取值都是取自 $f_1, f_2, \cdots, f_m$，因此，P 中各字段必然对应匹配于 R_i^{m-k} 中相应匹配域，所以对于 $P \to R_i (R_i \in \chi)$，则 $P \to R_i^{m-k} (R_i^{m-k} \in \chi^{m-k})$。

定义 2 对于经过裁剪后得到的匹配域（规则排列顺序相对于原规则集不变）χ^{m-k}，可能存在数据分组头部 P，使 P 在 χ 中没有匹配结果但在 χ^{m-k} 中能够得到匹配，此时 χ^{m-k} 中的匹配结果称为假阳性匹配。

以图 2 所示规则集为例，对于数据分组头部 $P = <1,5,3>$，原规则集中没有匹配结果，而对于提取 F_1 和 F_3 组成新的规则集，则返回匹配结果 R_1。此时，经过裁剪后的规则集输出了假阳性匹配结果。

对于假阳性结果，在实际硬件数据分组流水线中，可以通过添加一步检查步骤来消除假阳性结果带来的错误。对于裁剪后的规则集，可在 TCAM 输出匹配结果后增加一级 RAM，RAM 中存储每条规则中被裁剪掉的匹配域取值，根据 TCAM 输出匹配索引与数据分组头部中的相应字段作对比，若符合匹配，则输出匹配结果，否则丢弃匹配结果，将输出置为“无匹配”，如图 3 所示。由于在实际硬件环境上，相比于 TCAM，通常 RAM 资源较为丰富，因此增加的 RAM 资源消耗可忽略不计。

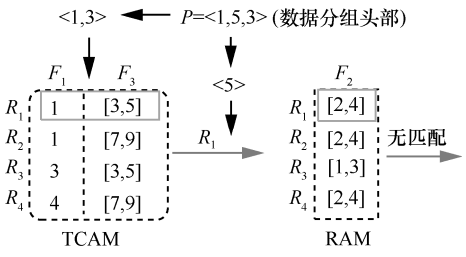


图 3 假阳性结果检验逻辑示意

实际中，一个规则集通常会设置“通配规则”并将其设置为最低优先权。当数据分组头部没有匹配到其他流表规则时，会匹配到“通配规则”，通配规则的相应处理动作即定义当规则集中无相应匹配规则时对数据分组的处理方式。本文中讨论的规则集在无特殊注明的情况下，默认不考虑“通配规则”。

3　基于信息熵的匹配域裁剪算法

上述分析可知，规则集经过适当的匹配域裁剪，可以有效减少匹配域数量，从而减小表项宽度，达到节省 TCAM 的目的。本节通过引入数据分组头部信息熵的概念，实现匹配域裁剪。

定义 3　对于所有数据分组头部，在经过匹配操作之前，其可能匹配于规则集中的任意一条规则。对于这种匹配结果的不确定性，称之为数据分组头部信息熵，记为 H 。

对于一个包含 N 条规则的规则集 χ，在匹配过程进行之前，一个数据分组头部可能有 N 种匹配结果，分别对应 N 条规则 $R_1 \sim R_N$ 。将规则 R_i 得到匹配的可能性记作 p_i，则此数据分组头部包含的信息熵为

$$H = -\sum_{i=1}^{N} p_i \mathrm{lb}\left(\frac{1}{p_i}\right) \tag{1}$$

在每条规则权重相同的情况下，有 $p_i = \frac{1}{N}$，则式(1)可简化为

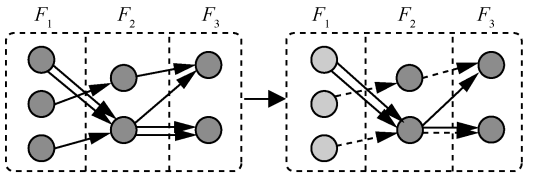
$$H = \mathrm{lb}N \tag{2}$$

在匹配域裁剪过程中，如第 2 节所述，需要分析每一个匹配域对规则集的区分度贡献作为匹配域裁剪的依据。其中，匹配域对规则集区分度的贡献可以通过该匹配域对数据分组头部信息熵的减少量来衡量，而数据分组的匹配过程，就是将数据分组头部信息熵降为 0 的过程。图 4(a)是图 2(a)所示规则集的图形化表示，从而方便以图论的方法来论述基于信息熵的规则集裁剪过程。该图以虚线划分为 3 个区域，分别对应匹配域 $F_1 \sim F_3$，每个节点代表一个匹配域取值，每条边对应一条规则在相应匹配域之间的取值。对于处于中间位置的匹配域，其每个节点在一条规则上向前和向后各有一条边连接到其他节点，为防止重复计算，将图中的边标为有向边，所有边的方向从前一个匹配域出发，指向后一个匹配域，接下来提到一个节点所连接到的边时，仅考虑从该节点出发的有向边（处于最后位置的匹配域中的节点连接到的边为指向该节点的有向边）。将一个规则集用图的形式表示后，每个节点所连接的边称为该节点的连接度，即该节点的度，记作 D ($D \leqslant N$)。

设一个匹配域 F_i 共包含 $V(F_i)$ 个不同取值，该匹配域中的节点 m 的度记为 $D_i(m)$ ($1 \leqslant m \leqslant V(F_i)$)，则单独使用该匹配域进行匹配操作后，数据分组头部信息熵减少为(N_x 为当前剩余规则总数)

$$H = H(\chi \mid F_i) = -\sum_{m=1}^{V(F_i)} \frac{D_i(m)}{N_x} \mathrm{lb}D_i(m) \tag{3}$$

以图 4(a)为例，单独 F_1 参与匹配时，数据分组头部信息熵降为 0.5 bit，单独 F_2 参与匹配时，数据分组头部信息熵降为 $0.75\mathrm{lb}3 \approx 1.18$ bit，单独 F_3 参与匹配时，数据分组头部信息熵降为 1 bit，由此可见，F_1 对分组头部信息熵降低贡献最大，优先选择 F_1；在 F_1 确定后，F_1 能够确定的规则即可从规则集中删除，即 F_1 中度为 1 的节点以及其所在规则可以从规则集中删除，如图 4(b)所示。图 4(b)中，F_1 节点颜色标记为浅灰色、虚线标记的规则表示从规则集中删除，相应地，F_2 和 F_3 中减掉与被删除规则的连接边。



(a)原始规则集图形化表示　(b)规则集特征分析图形化表示

图 4　规则集的图形化表示

在优先选择 F_1 后，剩下 2 条规则，使用 F_2 进行匹配操作后，分组头部信息熵为 1 bit，而使用 F_3 进行匹配后，分组头部信息熵减少为 0 bit，因此，选择 F_3 和 F_1 进行匹配查询即可消除全部数据分组头部信息熵。由此可见，使用分组头部信息熵进行匹配域裁剪时，其匹配域选择结果与第 2 节中的分析结果完全相同。

在上述分析中，通过信息上的对比对匹配域进行逐一提取，每提取一个匹配域字段只需进行 C_m^1 次信息熵的比较。通常，从所有匹配域中提取最优组合类似于集合覆盖问题（set cover problem），需要选取所有可能组合进行一一对比，从而得到最优组合，称为 NP 难问题。显然，若所有匹配域字段之间完全没有相关性，逐一提取得到的匹配域组合与遍历所有可能的组合得到的最优解相同。由于网络协议的设计特点，协议字段经常是成对存在的，如<源 IP 地址,目的 IP 地址>，下面着重讨论匹配字段关联性成对存在时逐一提取的效果。

定理 3　在匹配域字段之间相关性成对存在时，基于信息熵的匹配域逐一提取在提取相同数量的字段时能得到最优解。

证明 图 5 只表示出某规则集中的 2 个存在明显相关性的匹配域字段 F_1 和 F_2（规则集中其他匹配域并未在图中表示）。由于假设关联性成对存在，其他匹配域与 F_1、F_2 之间不存在明显关联性。

如图 5 所示，对于此规则集中的所有规则，在确定某条规则的匹配域 F_2 的取值后，则其相应的 F_1 的取值也得到确定。此时，F_1 和 F_2 之间具有强相关性，并且存在 F_2 到 F_1 的映射。此时，选择 F_2 匹配后的剩余分组头部信息熵 $H_2 \leqslant H_1$。对于存在强相关性的 2 个匹配域 F_1 和 F_2，记同时使用这 2 个匹配域匹配操作后的数据分组头部信息熵减少量为 $\Delta H_{1,2}$，分别使用 F_1 和 F_2 进行匹配操作后减少的数据分组头部信息熵减少量为 ΔH_1、ΔH_2，则有 $\Delta H_{1,2} \ll \Delta H_1 + \Delta H_2$。而图 5 中，在逐一提取匹配域时，由于 $H_1 > H_2$，F_2 为优先选取的匹配域。而剩余规则集在提取出 F_2 后，有 $\Delta H_1' \ll \Delta H_1$（$\Delta H_1$ 为未提取 F_2 时使用 F_1 进行匹配的分组头部信息熵减少量，$\Delta H_1'$ 为提取出 F_2 后使用 F_1 进行匹配减少的分组头部信息熵减少量）。此时，若存在 F_3，有 $\Delta H_1' < \Delta H_3'$，则逐一提取时 F_3 优先于 F_1 作为被选取匹配域，此时 $\Delta H_{1,2} = \Delta H_2 + \Delta H_1' < \Delta H_2 + \Delta H_3'$；若不存在 F_3 使 $\Delta H_2' < \Delta H_3'$，则继续选取 F_2，此时 $\Delta H_{1,2} = \Delta H_1 + \Delta H_2'$。上述 2 种情况中，逐一提取的结果都是最优结果。

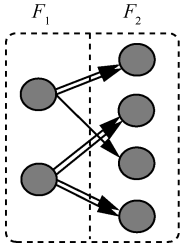


图 5 强相关匹配域对示意

4 硬件架构及算法实现

4.1 匹配域裁剪算法实现

如第 2 节所述，匹配域裁剪只能针对无序集进行。实际流表中，极少数规则之间可能存在重叠，算法 1 将规则集划分为无序集 G_I 和有序集 G_D，从而在后续步骤中，可以在 G_I 上执行匹配域裁剪。算法 1 将存在重叠的规则进行提取，每组重叠规则中，将优先权最高的一条规则（TCAM 中默认排序靠前者优先级高）存入 G_I 中，剩余规则按照相同的相对顺序存入 G_D。由于每条规则在最坏情况下需要在 m 个匹配域字段上分别进行对比，因此算法 1 在最坏情况下的复杂度为 $O(N^2m)$。而在实验中发现，在以 OpenFlow 为代表的规则集中，规则之间所需比较的平均字段数量要远小于 m。

算法 1 无序集搜索

输入 χ

输出 无序集 G_I，有序集 G_D

```
for i = 1 to N
  for j = (i+1) to N
    if R_i ∩ R_j ≠ ∅
      将 R_j 添加至 G_D;
      从 χ 中移除 R_j;
      将 R_i 添加至 G_I;
return G_I, G_D;
```

提取出规则集中的无序集后，即可对无序集 G_I 进行匹配域裁剪。实际流表的匹配域裁剪过程中，可能由于少部分流表匹配域取值的分散性，使兼顾此部分流表需要增加匹配域数量。当极少数流表的存在引起所需匹配域数量增多时，代价过大，此时放弃此部分流表，并将其从无序集 G_I 移到有序集 G_D 中（流表之间存放的相关顺序按照原规则集进行）。因此，设置覆盖参数 β $(0 \leqslant \beta \leqslant 1)$，当经过裁剪的匹配域组成的规则集能够覆盖的规则占总数量比例达到 β 时，即停止继续裁剪。具体步骤如算法 2 所示。在算法 2 中，默认各规则权重相同，因此参数 β 确定的规则数量占比同时也是数据分组头部原始信息熵的减少量。算法 2 遵循匹配域逐一提取的原则，在提取的匹配域字段能够区分大部分规则（比例达到 β 以上）时，算法终止，输出所提取的匹配域集合，并将剩余规则存入 G_D 中，输出更新后的 G_D。在最坏情况下，算法 2 在每次循环中需要遍历 m 个字段，对于每个字段计算信息熵时需要计算 $O(N)$ 个节点；并且对于外层循环，需要挑选出所有匹配域字段才能完成算法，此时复杂度为 $O(Nm^2)$。在通常情况下，由于算法 2 只需提取少数字段即可结束循环，算法实际执行复杂度要明显低于最坏情况下的复杂度。

算法 2 匹配域裁剪

输入 G_I，G_D，β

输出 S，G_D

```
将 S 初始化为 ∅
H = lb(N);  // N = |G_I|
```

while ($H > \mathrm{lb}(N(1-\beta))$) do

$i = \arg\min -\sum_{m=1}^{V(F_i)} \frac{D_i(m)}{N_x} \mathrm{lb} D_i(m)$;

$H = -\sum_{m=1}^{V(F_i)} \frac{D_i(m)}{N_x} \mathrm{lb} D_i(m)$;

添加 F_i 至 S;

删除能够被区分的规则;

更新 G_I;

$N_x = |G_\mathrm{I}|$;

将不能被区分的规则移至 G_D;

return S, G_D;

4.2　硬件架构设计

与软件算法相对应，基于信息熵的匹配域裁剪算法所需硬件架构如图 6 所示。在图 6 中，TCAM 芯片分为 2 个部分，一部分为小流表宽度的 N-TCAM，负责存储经过匹配域裁剪后的流表；另一部分为原始流表宽度的 W-TCAM，负责存储上述算法中有序集 G_D 中的流表，此部分表项没有经过裁剪处理。由于实际流表中，通常 G_D 所占比例很小，因此 N-TCAM 中存储了绝大部分表项。N-TCAM 的匹配结果，需要进行假阳性验证，RAM 2 即图 3 所述用以存储检验信息的 RAM。W-TCAM 和 N-TCAM 得到查表结果后，同时送往优先判决电路。由 G_I 和 G_D 生成过程可知，在同时得到匹配结果的情况下，G_I 中的表项优先级高于 G_D，优先判决电路输出 N-TCAM 的匹配结果，当且仅当 N-TCAM 无匹配结果时，输出 W-TCAM 匹配结果。为便于灵活配置、对不同规则集采取不同裁剪方案，图 6 中，RAM 1 用以存储所需提取的匹配域在分组头部域中的偏移位置，由匹配域提取电路根据 RAM 1 中的偏移信息从分组头部域中提取相应信息，同时送往 N-TCAM 和 W-TCAM 进行查表操作。

5　仿真验证

本文将基于信息熵的匹配域裁剪方案分别应用于两类流表进行仿真验证。一类是国家宽带网络与应用工程技术研究中心的 OpenFlow 协议测试流表，另一类是采用 ClassBench[16]产生的规则集。算法运行的硬件环境为 Intel Core i7-4790 3.6 GHz 处理器，8 GB 内存；仿真软件为 Matlab 2012a。

5.1　基于 OpenFlow 流表的性能测试

本文将基于信息熵的匹配域裁剪方案与 H-SOFT 进行压缩率对比，在不同流表规模下，实验结果如图 7 所示。由图 7 可以看出，在不同流表规模下，EC 的 TCAM 使用量明显小于 H-SOFT，约为 H-SOFT 的 60%，并且，随着流表规模的增长，EC 的压缩效果基本保持稳定。图 7 中，EC 的压缩率在覆盖参数 β=0.95 时要高于覆盖参数 β=0.99 时（压缩后与压缩前占用 TCAM 空前的比值越小，压缩率越高），这反映出参与实验的 OpenFlow 流表的信息熵在匹配域上分布比较集中，因此增加 W-TCAM 的存放量不利于减少 N-TCAM 中的匹配域使用量。

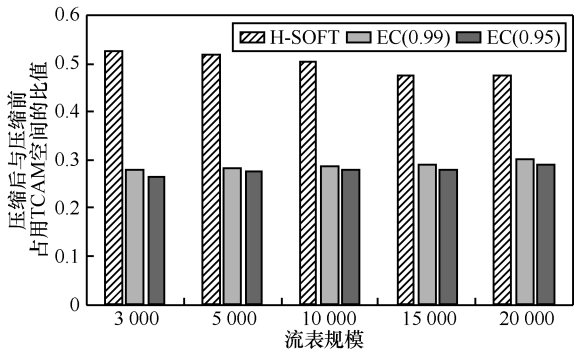


图 7　不同流表规模下 TCAM 压缩效率对比

H-SOFT 与 EC 的算法复杂度可以从算法运行时间上体现，如图 8 所示。由图 8 可以看出，在

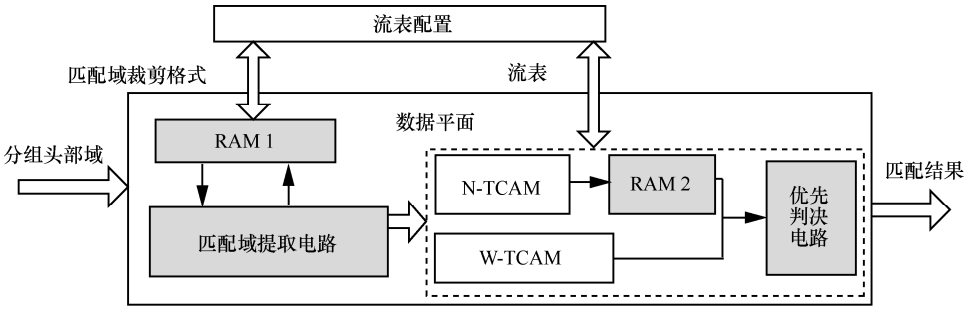


图 6　硬件逻辑架构

相同流表规模下，覆盖参数为 0.99 时 EC 的算法运行时间略多于 H-SOFT；并且，在实验数据范围内，二者的运行时间随流表规模增大而增长的速率近似。

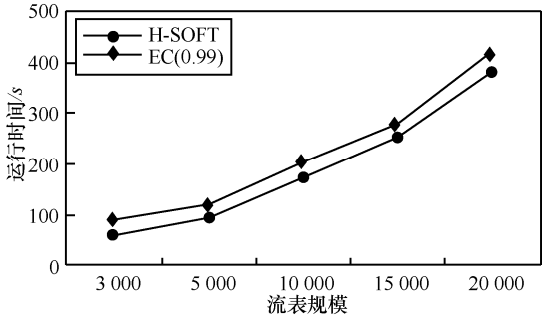


图 8 运行时间与流表规模关系

5.2 基于 ClassBench 规则集的性能测试

ClassBench 由华盛顿大学 Taylor 教授所开发，目前已经成为报文分类研究领域中广泛使用的规则集生成工具，能够生成 ACL(access control list)、FW(fire wall)、IPC(linux IP chains)3 类规则，并提供共 12 种参数模板。图 9 所示的 12 种规则集，由 ClassBench 提供的默认参数生成，每个规则集的规则数量为 10 000。由图中数据可以看出，大部分规则集只需提取 3 个匹配域即可完成匹配，少数规则集（如 FW4）需要 4 个匹配域。

文献[14]提出提取匹配域的 2-MGR 算法，在同样达到最优匹配域选取的前提下，其算法运行时间与 EC 对比如图 10 所示。可以看出，随着流表规模的增长，2-MGR 的算法运行时间急剧增长，而 EC 算法运行时间增长较为平缓，在流表规模达到 20 000 条时，2-MGR 的算法运行时间是 EC 的 3 倍左右。

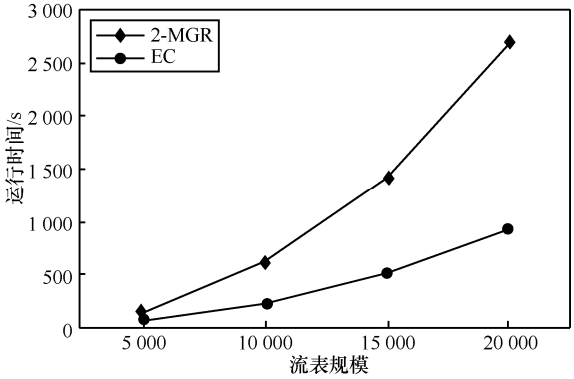


图 10 不同方案的算法运行时间

6 结束语

本文通过对分组分类规则集的匹配域取值分布进行分析，得到匹配域提取目标，减少了存储规则集所需 TCAM 表项宽度。同时，将数据分组匹配过程建模为分组头部信息熵在规则集中不断缩减的过程，通过从规则集到图的映射，在线性复杂度内完成了匹配域提取过程。实验结果表明，相对于现有的压缩方案，基于信息熵的匹配域裁剪算法 TCAM 存储空间压缩性能进一步提升了 40%，或随流表规模增长，算法运行时间能够明显减小。

参考文献：

[1] TAYLOR D E. Survey and taxonomy of packet classification techniques[J]. ACM Computing Surveys, 2005, 37(3):238-275.

[2] KANNAN K, BANERJEE S. Compact TCAM: flow entry compaction in TCAM for power aware SDN[C]// Distributed Computing and Networking. 2013:439-444.

[3] MCKEOWN N, ANDERSON T, BALAKRISHNAN H, et al. OpenFlow: enabling innovation in campus networks[J]. ACM Sigcomm Computer Communication Review, 2008, 38(2):69-74.

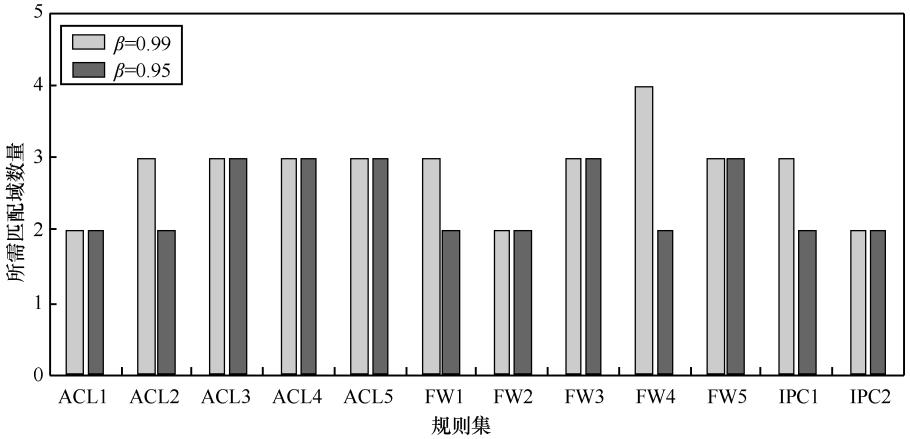


图 9 EC 在不同规则集上的匹配域提取结果

[4] LAKSHMINARAYANAN K, RANGARAJAN A, VENKATACHARY S. Algorithms for advanced packet classification with ternary CAMs[C]// ACM Sigcomm Computer Communication Review. 2005:193-204.

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

[6] BREMLER-BARR A, HAY D, HENDLER D. Layered interval codes for TCAM-based classification[C]//INFOCOM 2009. 2009: 1305-1313.

[7] BREMLER-BARR A, HARCHOL Y, HAY D, et al. Encoding short ranges in TCAM without expansion: efficient algorithm and applications[C]//ACM Symposium. 2016.

[8] MEINERS C R, LIU A X, TORNG E. Bit weaving: a non-prefix approach to compressing packet classifiers in TCAMs[J]. IEEE/ACM Transactions on Networking, 2012, 20(2):93-102.

[9] CHANG D Y, WANG P C. TCAM-based multi-match packet classification using multidimensional rule layering[J]. IEEE/ACM Transactions on Networking, 2016, 24(2): 1125-1138.

[10] MISHRA T, SAHNI S, SEETHARAMAN G. PC-DUOS: fast TCAM lookup and update for packet classifiers[J]. IEEE Symposium on Computers and Communications, 2011, 34(17):265-270.

[11] BANERJEE T, SAHNI S, SEETHARAMAN G. PC-DUOS+: a TCAM architecture for packet classifiers[J]. IEEE Transactions on Computers, 2014, 63(6):1527-1540.

[12] BANERJEE T, SAHNI S, SEETHARAMAN G. PC-TRIO: a power efficient TCAM architecture for packet classifiers[J]. IEEE Transactions on Computers, 2015, 64(4):1104-1118.

[13] GE J, CHEN Z, WU Y, et al. H-SOFT: a heuristic storage space optimisation algorithm for flow table of OpenFlow[J]. Concurrency & Computation Practice & Experience, 2014, 27(13):3497-3509.

[14] KOGAN K, NIKOLENKO S I, ROTTENSTREICH O, et al. Exploiting order independence for scalable and expressive packet classification[J]. IEEE/ACM Transactions on Networking, 2015(99):1-14.

[15] LIU Z, WANG X, YANG B, et al. BitCuts: towards fast packet classification for order-independent rules[J]. ACM SIGCOMM Computer Communication Review, 2015, 45(4): 339-340.

[16] TAYLOR D E, TURNER J S. ClassBench: a packet classification benchmark[J]. IEEE/ACM Transactions on Networking, 2005, 3(3): 499-511.

作者简介：

孙鹏浩（1992-），男，山东青岛人，国家数字交换系统工程技术研究中心硕士生，主要研究方向为可编程网络、数字交换技术。

兰巨龙（1962-），男，河北张北人，博士，国家数字交换系统工程技术研究中心总工程师、教授、博士生导师，主要研究方向为新一代信息网络关键理论与技术。

张少军（1989-），男，山西新绛人，国家数字交换系统工程技术研究中心博士生，主要研究方向为软件定义网络。

李军飞（1989-），男，河南安阳人，国家数字交换系统工程技术研究中心博士生，主要研究方向为软件定义网络。