第 38 卷 第 21 期
Vol.38 No.21

计 算 机 工 程
Computer Engineering

2012 年 11 月
November 2012

·开发研究与设计技术· 文章编号：1000—3428(2012)21—0283—03 文献标识码：A 中图分类号：TP393

一种基于 TCAM 的有效包分类方法

杨 迪

(中国科学技术大学苏州研究院计算机科学与技术学院，江苏 苏州 215000)

摘 要：基于三态内容可寻址内存(TCAM)的包分类方法不能有效解决区间膨胀的问题。为此，提出一种有效包分类方法。对包分类规则集中各个域的不同区间进行分组，利用 Shadow Encoding 方法对同一分组中的所有区间进行重新编码，依据重新编码的区间结果改写原始规则集。实验结果表明，该方法可以平均压缩 75.90%的 TCAM 存储空间。

关键词：包分类；三态内容可寻址内存；区间膨胀；区间分组；区间编码；流水线匹配

An Efficient Packet Classification Method Based on TCAM

YANG Di

(Department of Computer Science and Technology, Suzhou Institute for Advanced Study,
University of Science and Technology of China, Suzhou 215000, China)

【Abstract】Ternary Content Addressable Memory(TCAM)-based packet classification methods suffer from the range expansion problem. In this paper, an efficient range encoding method to solve this problem is proposed. It divides all the unique ranges in each field of rules into different groups, and uses the shadow encoding method to re-encode them, rewrites the original rule set according to re-encoding ranges. Experimental results show that this method can reduce TCAM storage space by 75.90% on average.

【Key words】packet classification; Ternary Content Addressable Memory(TCAM); range expansion; range grouping; range encoding; pipeline matching

DOI: 10.3969/j.issn.1000-3428.2012.21.075

1 概述

网络应用的普及衍生出很多新型的网络服务，如服务质量(Quality of Service, QoS)、网络数据包过滤、虚拟专网、流量计费等，而包分类正是所有这类应用的核心技术[1]。简单来说，包分类就是将数据包包头中的相关域内容同既定的规则集进行比较，进而决定对该数据包采取何种操作的过程。一条包分类规则由 5 个域构成，分别为源 IP、目的 IP、源端口、目的端口以及协议。

三态内容可寻址内存(Ternary Content Addressable Memory, TCAM)硬件是一种专门做并行比较的硬件，由于其具有快速的并行查找特性，因此 TCAM 被越来越多的应用在包分类领域当中。

因为 TCAM 仅支持存储前缀形式表示的规则条目，所以必须将规则转换成前缀形式才能用三态内容可寻址内存进行存储。如一条规则 R，其源端口和目的端口均包含区间[1,5]，而该区间，需要 3 条前缀表示，即“001”，“01*”和“10*”。因此，为了存储规则 R，TCAM 将需要使用 3×3=9 条三态内容可寻址内存条目，这就是所谓的“区间膨胀”问题。基于三态内容可寻址内存的包分类研究主要分 3 类：

(1)改进 TCAM 的硬件结构使之支持区间匹配[2]。

(2)在保证语义不变的前提下，对原始规则集的冗余条目进行压缩[3-4]。

(3)通过对区间进行重新编码来压缩存储空间[5-7]，而本文中提出的方法则属于第 3 类方法。

文献[5]提出对于规则集中的所有端口区间分配特定的比特进行表示，虽然可以完全解决“区间膨胀”问题，但是当区间个数较多的时候，该方法就不太切实可行。文献[6]提出一种基于上述方法的改进版本 DRES，根据各个端口区间出现的不同频率，对于出现频率较高的区间分配特定的比特进行编码。该选择性编码方法既可以有效地控制编码长度，同时，也能较好地压缩存储空间。文献[7]提出拓扑变换方法，该方法观察到域宽为 w 的区间中存在大量的相同区间，因此，可以将这些区间映射成为域宽远小于 w 的新区间，并且对于映射得到的新区间能用更少的前缀进行表示，但是其对于相互交叉的区间编码效果并不理想。针对这种情况，本文提出一种基于 TCAM 的包分类方法来解决“区间膨胀”问题。

作者简介：杨 迪(1987－)，男，硕士研究生，主研方向：网络技术，路由算法

收稿日期：2012-02-01 **修回日期：**2012-02-27 **E-mail：**yang.di.ustc@gmail.com

2 区间编码方法

本文提出的包分类算法通过区间分组和区间重编码等技术，在有效解决“区间膨胀”问题的同时，还可以减少包分类规则的位宽，因此，大大降低整个 TCAM 的存储空间开销。

本文提出一种区间分组算法，根据区间之间的相互关系，对规则集中各个域的不同区间进行分组，对同一个分组中的所有区间构建一颗关系树，接着利用 Shadow Encoding 方法[8]对该棵关系树中的所有区间进行重新编码，根据重新编码的区间结果，对原始规则集进行改写，得到最终的区间转换表(预处理作用)和规则分类表(并行匹配查找作用)。本文提出一种并行流水线匹配机制，使得可以在 T_{tcam} 时间内完成整个匹配流程。

2.1 区间分组

包分类规则集中的每条规则由数据包头部的 5 个域内容组成，需要对这 5 个域上的所有不同区间进行分组。首先对任意 2 个不同的区间 R_i 和 R_j 的关系进行如下定义：

(1)无关： $R_i \cap R_j = \varnothing$ 。

(2)子集： $R_i \subset R_j$ 。

(3)超集： $R_j \subset R_i$ 。

(4)交集： $R_i \cap R_j \neq \varnothing$ 。

区间分组算法如下：

Step1 找出第 i 个域上的“不同”区间 R_i，初始化分组数 $K_i = 0$ 。

Step2 遍历第 i 个域上的“不同”区间 R_i 。

Step3 遍历分组 G_m，其中， $m < K_i$ 。

Step4 遍历所有区间 R_j，其中， $R_j \in G_m$ 。

Step5 如果 R_i 和 R_j 为“交集”，转 Step3。

Step6 遍历完所有区间 R_j，将区间 R_i 加入分组 G_m 当中，转 Step2。

Step7 遍历完所有分组 G_m，新建分组 m'，将 R_i 加入 G_m 中，分组数 $K_i + 1$，转 Step2。

Step8 输出所有分组 G_m 。

2.2 区间编码

利用区间分组算法，将规则集单个域上的所有区间划分成了 K_i 个分组，每个分组的区间定义域为该单个域的取值空间。如 16 bit 的端口域，其取值空间为 $[0, 2^{16}-1]$ 。根据区间分组算法的分组策略，同一个分组中的任意 2 个区间 R_i 和 R_j 只存在 3 种关系，即无关、子集或超集。利用此种特性，对该分组中的所有区间构建一棵关系树，然后利用 Shadow Encoding 方法[8]进行编码。构建的关系树具有以下特点：

(1)树中的每个节点 N_i 对应分组中的一个区间 R_i 。

(2)树中的每个节点(除根节点外)，有且仅有一个父节点，可以有 0 或多个孩子节点。

(3)2 个节点 N_i 和 N_j，在满足 $R_i \subset R_j$ 的同时，不存在一节点 N_k，使得 $R_i \subset R_k$，且 $R_k \subset R_j$，则节点 N_j 是节点 N_i 的父节点。

(4)根节点 N_{root} 对应的区间必为该分组的区间定义域。

通过举例来简单说明该编码方案，假设存在一个分组 G，其中，包含 6 个区间，分别为[5,19]、[20,39]、[40,62]、[30,38]、[50,58]、[59,60]，该分组的区间定义域为[0,63]，因此，需要将区间[0,63]加入分组 G 中。

Shadow Encoding 方法对 7 个区间的编码过程如图 1 所示。

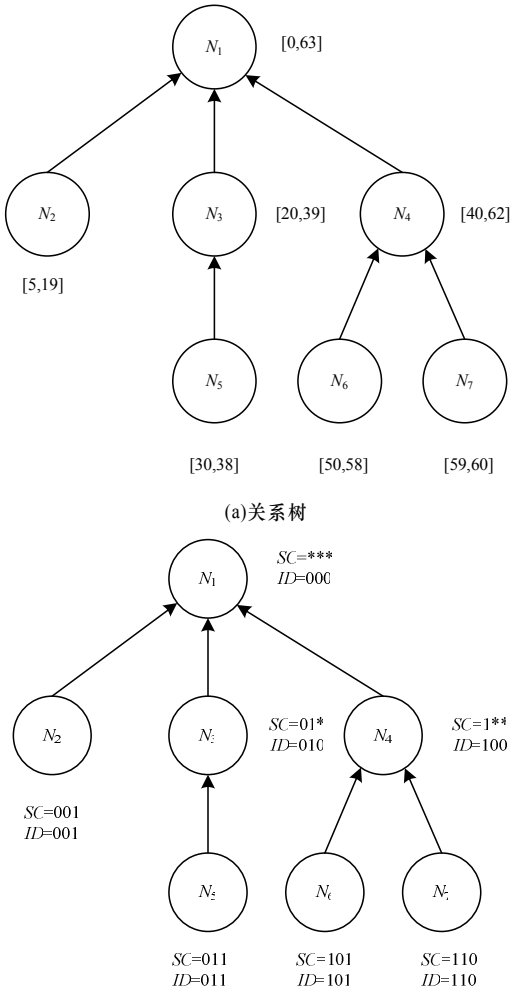

图 1 Shadow Encoding 方法对 7 个区间的编码过程

针对该 7 个区间，构建一颗关系树如图 1(a)所示。利用 Shadow Encoding 编码方案，对这 7 个区间进行重新编码，重新编码之后的结果如图 1(b)所示。编码结果包含 2 个部分，*SC* 码段，该编码用来辅助生成规则分类表；*ID* 码段，该编码用来辅助生成区间转换表。

2.3 规则分类表和区间转换表

规则分类表的生成方法如下：每条规则由所有 K 个分组的编码结果拼接而成。对规则中的第 i 个分组的编码段具体赋值操作为，如果该分组包含的区间内存在有原始规则定义的区间，则将该码段置为该区间经过 Shadow Encoding 编码方案重新编码的结果 *SC* 值，如果不存在这样的区间则将该码段置“全*”，那么对所有规则采用上述

方法进行重新改写，将改写之后得到的结果称之为规则分类表。

区间转换表的生成方法如下：首先把每个分组包含的所有区间划分成不相交的连续区间，然后将这些连续区间的转换结果赋值为在关系树中包含该区间，且深度最深的节点对应的 ID 值。如在图 1(b)所示的关系树中，区间[30,38]对应的区间转换结果为 $ID=011$。对于 K 个分组，需要 K 张区间转换表。为了达到压缩区间转换表空间的目的，可以将属于相同域的 K_i 张区间转换表，合并为一张区间转换表。具体操作方法为：将这 K_i 张区间转换表中，所有连续区间继续划分成不相交的连续区间，而新得到的每个连续区间在这 K_i 张区间转换表中对应的转换结果分别为 $ID_1, ID_2, \cdots, ID_{K_i}$，则该区间在合并之后的区间转换表中的结果即为 $ID_1 | ID_2 | \cdots | ID_{K_i}$，其中，|为连接操作符。

2.4　算法并行流水线匹配机制

在具体实现中，将源 IP 域划分成前后 2 个 16 bit 的新域，对目的 IP 域也采取相同的操作。因此，每条规则将由 7 个域构成，分别为源 IP_1(前 16 bit)，源 IP_2(后 16 bit)，目的 IP_1(前 16 bit)，目的 IP_2(后 16 bit)，源端口(16 bit)，目的端口(16 bit)和协议(8 bit)。根据规则的构成形式，最终将得到 1 张规则分类表和 7 张区间转换表。6 张区间转换表(除协议域外)将需要 6 个长度为 64 KB 的一维数组，而对于协议域的区间转换表，则需要一个大小为 256 Byte 的一维数组，因此，可将这 7 张区间转换表存储在 SRAM 当中。对于规则分类表，由于需要并行查找经过 7 张区间转换表预处理之后的内容，因此需要将规则分类表存储在 TCAM 中。

当需要分类一个数据包时，首先将该数据包头部 7 个域的内容经过 7 张区间转换表预处理，然后将 7 个预处理的结果拼接成一个新的查找关键词，并放入规则分类表中进行并行查找，算法匹配流程如图 2 所示。

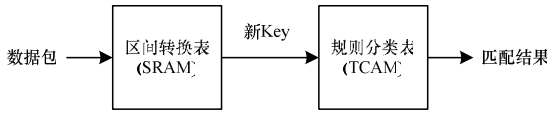

图 2　算法匹配流程

整个分类过程总共需要 7 次 SRAM 和一次 TCAM 查找操作，所需的分类时间为：$7\times T_{\text{sram}}+T_{\text{tcam}}$。为了加速整个分类速度，可以采用二级并行流水线的匹配机制，由于 7 张区间转换表的查找操作互不影响，因此 7 次 SRAM 的查找操作也可以同时进行。采取上述的匹配机制，整个分类时间即为 $\max(T_{\text{sram}}, T_{\text{tcam}})=T_{\text{tcam}}$。

3　实验结果与分析

本文利用 ClassBench[9]规则集种子文件生成 3 种类型的实验规则集，即访问控制列表(Access Control List, ACL)、防火墙(Firewall, FW)和 IP 链(IP Chain, IPC)。规则集的相关特性如表 1 所示，其中，膨胀率为 TCAM 数除以规则数。

表 1　规则集的相关特性

规则集	规则数	TCAM 数	膨胀率
acl-1	921	1 238	1.34
acl-2	907	1 996	2.20
fw-1	846	3 221	3.81
fw-2	979	1 874	1.91
ipc-1	953	1 309	1.37
ipc-2	719	719	1.00

针对这 6 个规则集，共实现 4 种方法，并从 TCAM 存储空间和匹配速度 2 个方面对其进行比较。上述 4 种方法分别是：将原始规则集直接存储在 TCAM 中(DIRC)，动态区间编码方法(DRES)，拓扑变换方法(TOPO)以及本文提出的区间分组法(RG)。TCAM 存储空间定义为所需的 TCAM 条数与每条规则实际位宽的乘积。

表 2 列出了 4 种包分类方法关于 TCAM 空间开销的实验结果。在所有 6 个包分类规则集上，本文方法可以平均压缩 75.90%的 TCAM 存储空间。

表 2　4 种包分类方法的 TCAM 空间开销实验结果　bit

规则集	DIRC	DRES	TOPO	本文方法
acl-1	128 752	124 335	45 159	34 998
acl-2	207 584	96 142	46 861	38 094
fw-1	334 984	91 368	38 952	34 686
fw-2	194 896	102 795	66 249	45 034
ipc-1	136 136	104 830	60 100	44 791
ipc-2	74 776	74 776	27 210	24 446

考虑到不同 TCAM 硬件工作频率的差别，采用包分类方法匹配单个数据包所需的 TCAM 工作频率 T_{tcam} 的个数表示包分类方法的匹配速度。通过使用并行流水线的匹配机制，以上 4 种包分类方法均可以在 T_{tcam} 时间内完成一个数据包的分类操作。综上，本文方法可以在不损失匹配速度的同时，所需的 TCAM 空间开销均优于其他方法。

4　结束语

本文提出一种基于 TCAM 的包分类方法。通过对包分类规则集中的区间进行适当分组及重新编码，解决区间膨胀问题。实验结果表明，该方法可以降低 TCAM 的存储空间。如何能高效地对具有交集关系的区间进行编码是今后的研究方向。

参考文献

[1] 翟　钰, 武舒凡, 胡建武. 防火墙包过滤技术发展研究[J]. 计算机应用研究, 2004, 21(9): 144-146.

[2] Spitznagel E D T, Turner J. Packet Classification Using Extended TCAMs[C]//Proc. of the 11th IEEE International Conference on Network Protocols. Washington D. C., USA: IEEE Computer Society, 2003.

[3] Meiners C R, Alex L, Torng E. TCAM Razor: A Systematic Approach Towards Minimizing Packet Classifiers in TCAMs[J]. IEEE/ACM Transactions on Networking, 2010, 18(2): 490-500.

(下转第 289 页)