

# Survey and Taxonomy of Packet Classification Techniques

**Table I.** Example Filter Set of 16 Filters Classifying on Four Fields  
(Each filter has an associated flow identifier (*Flow ID*) and priority tag (*PT*) where † denotes a nonexclusive filter; wildcard fields are denoted with \*)

Filter				Action	
SA	DA	Prot	DP	FlowID	PT
11010010	*	TCP	[3:15]	0	3
10011100	*	*	[1:1]	1	5
101101*	001110*	*	[0:15]	2	8 <sup>†</sup>
10011100	01101010	UDP	[5:5]	3	2
*	*	ICMP	[0:15]	4	9 <sup>†</sup>
100111*	011010*	*	[3:15]	5	6 <sup>†</sup>
10010011	*	TCP	[3:15]	6	3
*	*	UDP	[3:15]	7	9 <sup>†</sup>
11101100	01111010	*	[0:15]	8	2
111010*	01011000	UDP	[6:6]	9	2
100110*	11011000	UDP	[0:15]	10	2
010110*	11011000	UDP	[0:15]	11	2
01110010	*	TCP	[3:15]	12	4 <sup>†</sup>
10011100	01101010	TCP	[0:1]	13	3
01110010	*	*	[3:3]	14	3
100111*	011010*	UDP	[1:1]	15	4

## 1. Introduction

### 1.1 Constraints

- 时间复杂度
- 针对病态研究 (worst case)
- 规则集增长而扩展
- 功耗

### 1.2 Avoiding the Problem

- avoid performing IP lookups and packet classification via emulated circuit switching
- 多个AS之间的协调

## 2. 过滤器集的特征

- 当前 (2005) filter set比较小 (少于5000)
- 协议种类很少 (TCP, UDP, wildcard) others (ICMP, IGMP, (E)IGRP, GRE and IPIP.)
- 传输层协议端口范围大 (>1023) -> 前缀转化效率低
- 匹配给定地址的唯一地址前缀的数量通常不超过5个。
- 分组匹配过滤数小于5个
- 不同过滤器共享许多匹配条件
- 过滤器集合中指定的唯一匹配条件的数量明显少于过滤器集合中的过滤器数量。

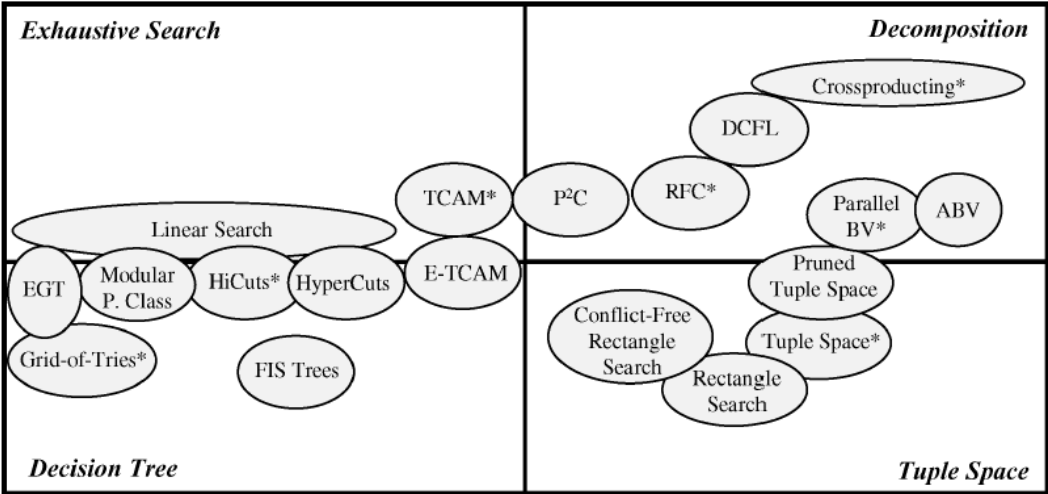
总结重复度较小，一个字段匹配上后可以显著减少可匹配filters的数量

通常两个字段可以完成匹配

### 3. 方法分类

根据问题采用的数据结构进行分类。

- 穷举法：测试所有entries
- 决策树：
- 分解（分治）
- 元组空间：感觉类似桶排序



**Fig. 1.** Taxonomy of multiple field search techniques for packet classification; adjacent techniques are related; hybrid techniques overlap quadrant boundaries; \* denotes a seminal technique.

介绍算法：

描述算法

提取问题规模

仿真结果

强化基本性能要求

### 4. 穷举法

线性搜索

集合并行搜索（TCAM）

吞吐量和并行数成正比

计算能力和并行数成正比

空间存储O(N)

IPv4 5-tuple -> 168bits(SIP:32 DIP:32 SP:16 DP:16 protocol: 8,以及分别32位的mask)

可以采用比特压缩技术，压缩到=>对数级别

## 4.1 线性搜索

$O(N)$ 级别存储,  $O(N)$ 访存, pipelines减少到 $1/P$ ,  $P$  is the pipelining stage

常用于查询最后一步

## 4.2 TCAM

缺陷:

- high cost
- 存储效率低下
- 能源消耗大
- 有限的扩展

访存次数大、3\times SRAM

E-TCAM实现范围匹配

## 5. 决策树

---

最长前缀匹配

任意范围匹配

精确匹配

=>一个规则存储在多个叶子节点 (规则复制问题)

Cutting

深度有界=>pipeline 结构大幅提高速度

### 5.1 Grid of tree

没有完全消除节点复制问题

Gridoftries 采用最长前缀匹配, 对多领域匹配不够好

### 5.2 EGT

改为跳转指针可多个字段搜索。

### 5.3 HiCuts

### 5.4 Modular Packet Classification

索引跳转表、搜索树和过滤桶。

先选 $k$ 位做桶, 然后再从每个桶里找

### 5.3 HyperCuts

### 5.6 Extended TCAM

没看懂

Cuts完的TCAM?

## 5.7 Fat Inverted Segment (FIS) Trees

不容易支持高速的动态更新

## 6. Decomposition

分解多字段问题：聚合

有点：并行

### 6.1 BV

bit 表示当前区域覆盖的Filter，后取交集

Survey and Taxonomy of Packet Classification

**Table IV.** Example Filter Set  
(Address field is 4-bits and port ranges cover 4-bit port numbers.)

Filter	Address	Port
<i>a</i>	1010	2 : 2
<i>b</i>	1100	5 : 5
<i>c</i>	0101	8 : 8
<i>d</i>	*	6 : 6
<i>e</i>	111*	0 : 15
<i>f</i>	001*	9 : 15
<i>g</i>	00*	0 : 4
<i>h</i>	0*	0 : 3
<i>i</i>	0110	0 : 15
<i>j</i>	1*	7 : 15
<i>k</i>	0*	11 : 11

### 6.2 Aggregated Bit-Vector (ABV)

A 个箱子，每个箱子只检索bit为1

将BV分成A块，每块只要有一个bit为1，则对应块记为1，加速查找

### 6.3 Crossproducting

叉乘，再查表

### 6.4 递归流分类（RFC）

高查找速度，但是浪费空间

RFC执行一个多阶段缩减到最终eqID，该eqID指定了应用到数据包的动作。

```
int count = 0x0000;
for(auto set : sets)
{
    if(set not in RFC)
    {
        RFC.insert(set);
        map[set] = count;
        count++;
    }
}
```

补一下查找

6.5 Parallel Packet Classification (P2C)

编码和中间结果聚合

参考最短选择木棍贪心算法，把每个有重叠的filter拿出来建立一层layer，相应的layer建立vector bits=>构造3元匹配字段（间隔、layer、vector bits）

层数等于任何端口号重叠范围的最大数目。

6.6 Distributed Crossproducting of Field Labels (DCFL)

7. 元组空间

Table V. Example Filter Set  
(Address fields are 4-bits and port ranges cover 4-bit portnumbers.)

Filter	SA	DA	SP	DP	Prot	Tuple
a	0*	001*	2 : 2	0 : 15	TCP	[1, 3, 2, 0, 1]
b	01*	0*	0 : 15	0 : 4	UDP	[2, 1, 0, 1, 1]
c	0110	0011	0 : 4	5 : 15	TCP	[4, 4, 1, 1, 1]
d	1100	*	5 : 15	2 : 2	UDP	[4, 0, 1, 2, 1]
e	1*	110*	2 : 2	0 : 15	UDP	[1, 3, 2, 0, 1]
f	10*	1*	0 : 15	0 : 4	TCP	[2, 1, 0, 1, 1]
g	1001	1100	0 : 4	5 : 15	UDP	[4, 4, 1, 1, 1]
h	0011	*	5 : 15	2 : 2	TCP	[4, 0, 1, 2, 1]
i	0*	110*	2 : 2	0 : 15	UDP	[1, 3, 2, 0, 1]
j	10*	0*	2 : 2	2 : 2	TCP	[2, 1, 2, 2, 1]
k	0110	1100	0 : 15	0 : 15	ICMP	[4, 4, 0, 0, 1]
l	1110	*	2 : 2	0 : 15	*	[4, 0, 2, 0, 0]

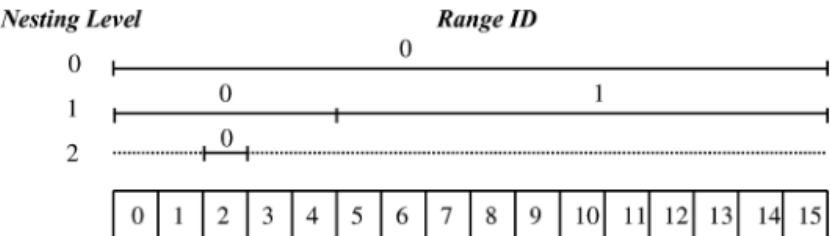


Fig. 21. Example of assigning tuple values for ranges based on Nesting Level and Range ID.

给定端口字段上的所有范围都放在一个不重叠的层次结构中，类似P2C，每个field为一个tuple，每个tuple都不带重复的，每个tuple为对应field的前缀长。

## 7.1 元组空间查找与元组剪枝

基本方案，7个元组一起探测

剪枝方案，分别探测每个元组后取交集

## 7.2 矩形查找

将元组分桶，但是如果field稍微多一点点，这个桶会爆炸？

然后从大往小（最长前缀开始搜索单元格）

## 7.3 Conflict-Free Rectangle Search

## 8.caching

---

使用cache 并不能解决最差情况

高速链路上的信息流缺乏引用的局域性