

基于多域并行编码的高速 IPv6 流分类

李 维, 刘 斌, 郝 颖, 林 伟, 唐 毅

(清华大学计算机科学与技术系, 北京 100084)

摘 要: IPv6 的多域流分类是高速路由器设计中的一个难点. 本文提出了一种使用 TCAM 的高速 IPv6 流分类方案, 其核心思想是: (1) 区分 IPv6 包头 5 个域字段的不同特征, 根据 IPv6 地址的特征及其分配信息对其进行压缩, 对 TCP 端口域实施扩展的层次编码, 根据统计数据对协议域进行压缩, 最终结果是把原始域的 296 比特转换成 280 比特的查找关键字, 与 TCAM 的表项宽度相匹配; (2) 使用嵌入 SSRAM 表查找技术, 对 5 个域并行进行独立编码, 消除瓶颈编码环节, 达到线速处理要求; (3) 分类规则数据库按照本文预设计的编码方式存储在 TCAM 中, 使用流水线技术让域的编码操作和查找操作并行执行, 每个 TCAM 访问周期完成一次查找操作. 同时, 为解决范围匹配问题, 本文设计了一种预定义位宽的动态范围编码算法, 既节省了 TCAM 的存储空间, 又提高了硬件规则库的更新速度. 分析和仿真表明, 当路由查找和流分类共用一个 TCAM 时, 使用较低的工作频率(66MHz), 流分类和路由查找速度均可达到 22Mpps, 满足高速 OC2192 接口的线速查找与流分类要求.

关键词: IP 包; IPv6; TCAM; 多域流分类; 编码

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2007) 05-0972-06

Ultra2High Speed IPv6 Packet Classification Based on Parallel Multi2Field Encoding

LI Wei, LIU Bin, XI Ying, LIN Wei, TANG Yi

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract: One challenging issue in IPv6 high speed routers design is the multi2field classification at line2rate for interface OC2192 and beyond. This paper proposes an efficient ultra2high speed IPv6 packet classification engine (classifier) based on TCAM (Ternary Content Addressable Memory). The key ideas are: (1) encoding the five fields according to their distinct features: we compress the IPv6 source/destination address fields based on the features of their formats and the distribution styles, apply an extended hierarchical coding method to TCP source/destination port fields and encode the protocol field based on the statistics. In this way, the original five2tuple with 296bit length is converted into a 280bit one which can fit well into the configured width of existing commercial TCAMs; (2) an embedded SSRAM table search technique is adopted to independently encode the five fields in parallel, eliminating the encoding bottleneck and guaranteeing the line2rate processing; (3) The rules of the classifier are stored in TCAMs according to the pre2designed coding formats. Utilization of pipeline makes the field coding operation and the rules search running parallel, which results the classifier outputting a search result within a single TCAM access. What's more, to effectively solve the range matching problem, we present a dynamic range encoding scheme which greatly reduces the storage requirement for TCAMs and increases the updating speed. Analysis and simulation demonstrate that running TCAMs at a frequency of 66MHZ(relatively low), the engine can reach a speed of over 22M packets per second (Mpps) for both the lookup and the packet classification respectively when integrated, and this perfectly satisfies the OC2192 interface's line2rate processing.

Key words: IP packet; IPv6; TCAM; multi2field packet classification; coding

1 引言

近年来一些新兴的网络应用, 如防火墙、基于策略的路由、区分服务、QoS、流量计费等, 要求路由器能够支持 IPv6 流分类(packet classification)功能. Internet 骨干网

传输链路速率的快速提升和 IPv6 地址长度的增加对在高速路由器中实现 OC2192 接口的/线速 IPv6 流分类带来了挑战.

通常, IPv6 基于五个域(五元组)^[1,2], 即源/目的 IPv6 地址、源/目的端口号和协议号来实施流分类. 近年

来, 专用并行查找器件 TCAM(Ternary Content Addressable Memory) 被广泛用到流分类引擎的设计中, 其优点是: 查找速度快、实现简单; 缺点是: (1) 单位比特的 TCAM 更昂贵, 且存储芯片的容量相对较小; (2) 功耗大. 此外, 在工程实现方面 TCAM 还存在一些其它的不理想之处: (1) 现有的 TCAM 器件的存储宽度通常都配置为 36/ 72/ 144/ 288/ 576 比特^X, 而 IPv6 数据包的五元组总宽度达 296 比特, 不能直接匹配. 因此, 我们需要寻找一种新的方法来有效地压缩 IPv6 数据包的五元组的比特数; (2) 源/ 目的端口号有用范围(range) 来表示, TCAM 来实现范围匹配时, 一个范围可能需要转化为多条前缀, 使 TCAM 中的物理表项数大幅膨胀. 因此, 如何有效地支持规则的范围匹配也是一个关键问题.

Van Lunteren 提出了一种并行分组分类(Parallel Packet Classification: P2C) 方案^[1], 采用并行结构先分别在多个 SRAM 中进行不同域的编码, 将编码结果汇总, 然后在 TCAM 中进行编码的查找. 它解决了范围匹配造成的表项扩张和按域分割带来的规则重叠问题, 但是对于 IPv6 地址的编码没有提出快速算法. 另外, 在更新方面, 由于它对范围匹配采用了动态追加比特的分层编码方式, 极端情况下可能会引发大量的 TCAM 表项更新, 这将降低更新的性能.

为解决上述问题, 本文设计了一个支持增量更新的高速 IPv6 流分类器. 基于对五元组的有效编码算法, 采用了一种并行的结构, 流水作业时, 每次 TCAM 访存可得到一个查找结果. 对 IPv6 地址和协议域进行了压缩, 用预定义位宽的分层编码方法解决了范围匹配带来的 TCAM 表项扩张问题, 降低了存储需求, 并提高了更新速率. 分析和试验结果表明该方法完全胜任 OC2 192 接口的/ 线速 0 IPv6 流分类要求. 当使用 CYPRESS 10512 TCAM 级联模式时, 该分类器可支持 1. 8M 条规则.

2 域的编码方法

域编码将解决以下两个问题: (1) 把 IPv6 原始五元组的 296 比特压缩到 280 比特; (2) 由于范围查找带来的 2~ 6 倍的表项扩张问题. 本节将详细讨论不同域的压缩方案和对相应域的编码策略.

2.1 协议域压缩

常用的协议种类很少, 一般只有 10 种^[2, 7], 故将 8 比特的协议号用 4 比特编码表示.

2.1.2 IPv6 地址压缩

针对 IPv6 地址的压缩, 考虑到最长前缀匹配原则, 需要保证压缩后的编码依然具备原前缀的包含特征.

IPv6 地址分为三类^[9]: 多播地址, 单播地址和泛播地址.

(1) IPv6 协议中多播地址格式如图 1 所示, 其前 8 比特固定为全 1/ 0, 无实际意义^[9]. 因此, 可将多播地址的前 8 比特直接去掉, 这样组播地址从 128 比特被压缩到了 120 比特.



图 1 多播地址格式

(2) 单播地址格式如图 2 所示, 其中 TLA 是顶级集聚标识符, 由 IANA 来管理. 据最新的 IPv6 单播地址分配情况^[10], IPv6 地址的前 16 比特只有 4 种组合: 2001, 2002, 2003 和 3FFE. 为保证将来的扩展性, 我们使用 8 比特, 留出了大量的可扩展空间. 这样单播地址从 128 比特压缩到了 120 比特.

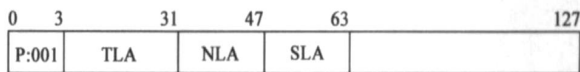


图 2 单播地址格式

(3) 由于泛播地址和单播地址前 64 位的表示格式相同, 因此可用与单播地址相同的方法将其压缩到 120 比特.

综上, 规则的源/ 目的地址无论是多播、单播还是泛播地址, 都可以压缩至 120 比特.

2.1.3 端口号编码

2.1.3.1 编码原则

端口范围有五种类型^[7]: 全范围(WC), 临时用户端口范围(HI[1024: 65535]:), 常用系统端口范围(LO[0: 1023]:), 精确端口号(EM) 和任意端口范围(AR). 考虑到精确端口号也是左右端点相等的范围, 因此把端口匹配统称为范围匹配. 为解决重叠和表项扩张问题, 本文使用一种预定义位宽的动态分层范围编码方法来消除范围重叠. 如果同一种类型端口范围之间存在重叠, 通过追加新层来把重叠的端口范围放在不同的层, 但每层分配的最大比特数是预先定义好的. 这种分层编码结构用 $1/ m/ n/$, 来表示, (m, n 分别标识各层分配的比特数) 其中 // 0 代表层. 具体编码原则如下:

(1) 使用 t (t 是正整数) 个比特来编码端口号, $t > = 1 + m + n$, 初始化所有比特为零, 并初始分配 3 层;

(2) 因为 HI 和 LO 互不重叠, 可放于同一层, 因此在第一层分配 1 比特用于表示 HI 和 LO;

(3) EM 类型的端口范围彼此不会出现重叠, 一层可以容纳全部的 EM, 故在第二层分配 m 比特来存放所有的 EM 端口号;

(4) 因为 AR 类型的端口范围之间会出现重叠, 第三层使用 n 比特来存放绝大多数不重叠的 AR;

X 现在也可以配置成 $(288 + 32N)$ 比特的宽度 (N 为有一定范围的自然数)^[3, 4]

(5) 对由于重叠而不能都放置于同一层的 AR, 可动态添加新的层来容纳它们。

下面举例说明编码原则如何应用. 对于一个初始包含 7 个端口规则的集合(HI, LO, 5000, 5001, [80: 81], [160: 161], [5000: 5001]), 根据上述分层动态编码原则, 总体分配 8 比特, 初始化分层结构为 1/2/3, 如图 3 所示.

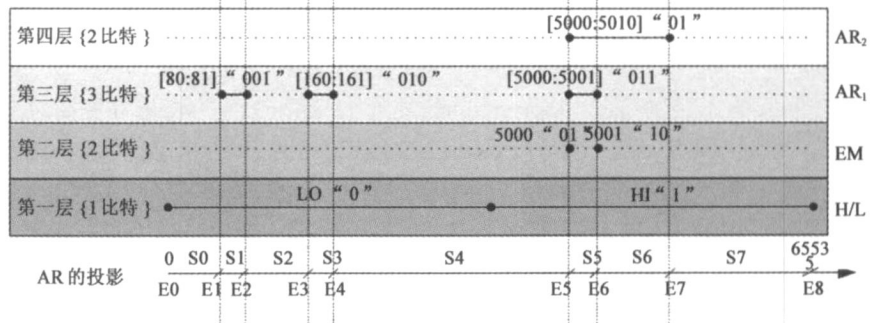


图 3 分层编码结构示例

21312 端口子编码与 TCAM 中端口规则的存储方案

每个到达的输入端口号都先进行/ 端口编码0, 并存在与之匹配的存储在 TCAM 中的/ 端口规则0. 因此分层编码会产生两种存储格式(端口编码和端口规则) 的字符串, 其格式和用途均不同, 具体如下:

(1) 端口编码是一个二值编码字符串. 根据 2.3.1 所述分层编码原则可知, 一个端口编码由相应的 3 个端口子编码(HI/ LO, EM 和 AR 的二值子编码) 构成. HI/ LO 和 EM 的二值子编码就是其在对应层上所分得的编码, 例如, 图 3 中 LO 和端口 5000 的二值子编码分别为/00 和/010. 对于 AR, 由于不同 AR 之间可能产生重叠, 所以需要先将所有 AR 投影至一数轴上, 这时存在重叠的不同区间将转换为一组头尾相接的原子区间, 原子区间中任意两个都不交叠. 图 3 中用 $S_i (i= 1, 2, M)$ 来表示原子区间, 原子区间的端点用 $E_i (j= 1, 2, M+ 1)$ 来表示, 其中 M 是原子区间的个数. 这样将每个任意端口范围 AR 转化为了一个或一组独立的 S_i , 因此在这个分层结构中只需要存储 S_i 的二值子编码, 即 S_i 在所有 AR 层的投影所对应的编码的顺序连接. 下面举例说明, 把图 3 的端口范围集合中所有的 AR 投影到一数轴上, 就得到了 S_0 至 S_7 这 6 个区间(分别为[0: 79], [80: 81], [82: 159], [160: 161], [162: 4999], [5000: 5001], [5002: 5010], [5010: 65535]) 以及区间的端点 E_0 至 E_8 按照上述规则可以得到 S_1 至 S_6 (S_0 和 S_7 没有实际意义, 故未列出) 的二值子编码字符串(见表 1), 例如 S_5 的二值子编码为/011010.

(2) TCAM 中存储的端口规则是三态字符串, 用于关键字重组后进行的流分类查找, 其存储顺序依据整条规则的优先级顺序. 由于一个端口范围只可能在特定的一层, 则一个端口范围的三态字符串的构成是: 该端口所在层对应的比特为该端口的编码, 其他比特为/* 0. 特殊地, WC 的三态字符串为全/* 0. 注意, 对于范围来说, 在 TCAM 中存储的是 AR(而不是其转化后的原子区间) 的三态字符串. 图 3 中各个端口规则的三态字符串如表 1 所示. 例如, LO 和 [5000: 5001] 的三态字符串分别为/0***** 0 和/ *** 01* * 0.

表 1 动态分层编码的二值字符串和三态字符串

端口范围	端口子编码 (二值字符串)	端口范围	端口规则 (三态字符串)
LO	0#####	WC	@ @ @ @ @ @ @ @
HI	1#####	LO	0 @ @ @ @ @ @ @
5000	# 0 1 #####	HI	1 @ @ @ @ @ @ @ @
5001	# 1 0 #####	5000	@ 0 1 @ @ @ @ @ @ @
S_1	### 0 0 1 0 0	5001	@ 1 0 @ @ @ @ @ @ @
S_2	### 0 0 0 0 0	[80: 81]	@ @ @ 0 0 1 @ @
S_3	### 0 1 0 0 0	[160: 191]	@ @ @ @ 0 1 0 @ @
S_4	### 0 0 0 0 0	[5000: 5001]	@ @ @ @ 0 1 1 @ @
S_5	### 0 1 1 0 1	[5000: 5010]	@ @ @ @ @ @ @ 0 1
S_6	### 0 0 0 0 1		

21313 端口号的端口编码的查找及其在 TCAM 中的匹配方式

查找时, 对输入端口号, 其端口编码即二值编码字符串由 HI/ LO, EM 和 AR 三种子查找得到的端口子编码按顺序串连组成(如图 4 所示), 如果在子查找中没有匹配, 则默认子编码为全零.

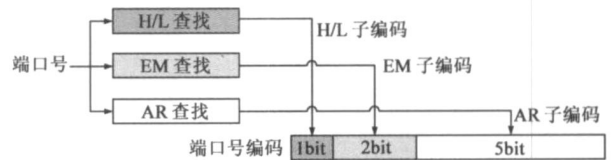


图 4 端口号编码的形成过程

21314 确定端口号编码参数

上两节介绍了编码原则和编码的获得及存储方案, 接下来需要根据实际情况分析已得到编码原则中合适的 t, m 和 n 的取值, 并论证编码的分配是充裕的.

根据对多个实际数据库的统计^[6,7], 得到:

- (1) 规则库中不同端口表示的数目远少于规则个数;
- (2) WC 和 EM 类规则占很大比例, AR 类规则占的比例很小, 源、目端口范围的统计规律有所不同, 故最终的处理方式不同;
- (3) 在 AR 类规则中, 相互重叠的 AR 个数不超过 4;
- (4) 现有规则库中, 源、目端口号中不同的 EM 个数分别不超过 27 和 183, 源、目端口号中不同的 AR 个数

更少, 分别不超过 5 和 50.

表 2 SP 与 DP 的统计数据

规则库类型	最大的规则库的大小	不同的 SP 和 DP 的个数		编码的分配		总共使用的编码位数	
		SP	DP	SP	DP	SP	DP
ACL	4557	3	235	1/2/	1/8/2/1/	3	12
FW	283	49	9	1/4/1	1/6/2/	6	9
IPC	1702	34	54	1/4/2	1/6/1/1/	7	9

模拟动态分层编码方案时, 使用了文献[12, 13]提供的 Classbench 生成了 12 个规则库. 这 12 个规则库包含了 3 种类型, 即访问控制列表(Access Control List: ACL), 防火墙(Firewall: FW)和 IP 链(IP Chain: IPC). 把动态分层编码方案应用到这 12 个规则库上, 得到如表 2 所示的 SP 和 DP 的分层编码. 可以看出 SP 和 DP 需要的编码比特数分别不超过 7 和 12, 故给 SP 和 DP 分别分配 12 和 24 比特的编码是充裕的. 同时结合表中各层的比特分配情况, 本文给出 SP 和 DP 的初始分层结构分别为: 1/6/2 和 1/9/6/2/, 这留下了较大的可扩展空间, 实际应用时可扩展到: 1/6/2/2/1/ 和 1/9/6/2/2/2/2/.

3 IPv6 流分类器的结构和查找算法

分类器采用了两级查找结构, 如图 5 所示. 第一级采用域间并行查找分别获得 IP 包各域的编码, 本文称为独立域查找; 第二级将各域的编码组合为一个新的关键字送入 TCAM, 得到流分类结果, 本文称为流分类查找.

本节主要论述独立域查找的结构和实现, 包括: 协议域, 源/目的 IPv6 地址和源/目的端口号.

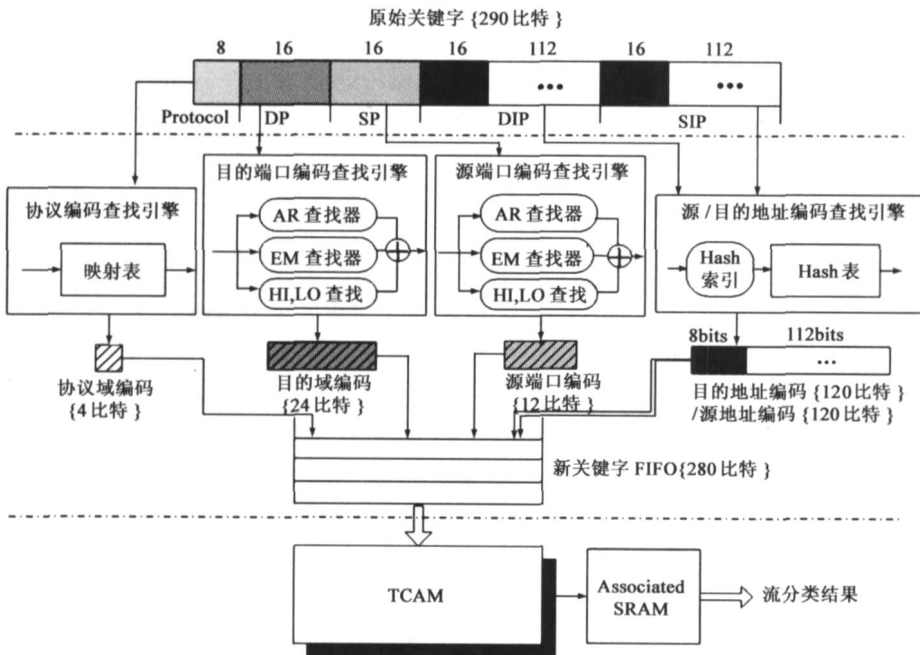


图 5 IPv6 流分类器的结构

311 协议域查找

协议域及其编码的长度较小, 分别为 8 比特和 4 比特, 使用一个简单的映射表(见图 5 中的协议编码查找引擎), 1 次 SRAM 访存周期即可查到结果.

312 IPv6 地址域查找

IPv6 地址前 16 比特的低 k(k 为正整数, 本方案中设为 8)位作为哈希索引进行查表. 得到压缩后的 8 比特的编码, 与 IPv6 地址后 112 比特组合成一个 120 比特宽的新编码值. 取 $k=8$, 每个哈希单元中的实际表项数不会超过 $1^{[10]}$. 使用一个 64 比特宽的 SRAM 来存储哈希表, 一个哈希单元(逻辑宽度 64 比特)可以容纳 4 条表项. 每条表项由端口号的高 8 比特和端口对应的 8 比特宽的编码组成. 一次访存可以读出多条表项并行比较, 这样 1 次 SRAM 访存(流水情况下 1 个 SRAM 周期)即可查到结果.

313 端口域查找

(1)EM(Exact Match) 查找

使用低 k 位作为哈希索引进行查表(统计得到, 对 SP, k 取 5; 对于 DP; k 取 6, 可以把所有 EM 比较均匀地分配到各个哈希单元中). 经对多个数据库进行统计分析, 一个哈希单元中表项条数不会超过 4, 故取一个哈希单元逻辑宽度为 144 比特, 包含 5 条表项, 每条表项由 16 比特的端口号和 $m(m < 9)$ 比特的 EM 的编码组成. 1 次访存读出 5 条表项, 1 次 SRAM 访存(流水情况下 1 个 SRAM 周期)便可查到 EM 的编码结果.

(2)AR(Arbitrary Range) 查找

对图 3 中的端口集合的例子, 用一个 T(这里 $T=3$)叉分段树^[4]来存储相应的数据结构, 用 SRAM 来存储, 图 6 所示. 最多 $\log_2 M$ 查找时间, 最多 $2M/t$ SRAM 更新时间, $(T@d_1 + (T-1)@d)@(M-1)/(T-1)$ 的 SRAM 存储空间, 其中 d 是端口宽度; d_1 是指针宽度, 在本设计中都取为 16 比特. 经统计分析, 取 $t=5$, 这样对于有 60 个 AR(实际规则库中 AR 的个数很少, 据文献[7]对 12 个实际规则库的统计, 最多的也仅有 50 个 AR)的规则库来说, 最多需要 3 次 SRAM 访存(流水情况下 6 个 SRAM 周期)可查出编码值.

4 更新算法

规则库的更新包括 3 个部

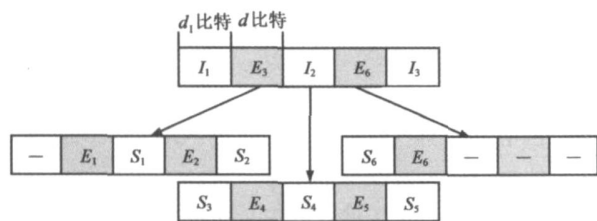


图 6 二叉分段树的结构示例

分:(1)域编码的更新;(2)相应 SRAM 中独立域查找结构的更新;(3)TCAM 内容的更新.后两种更新可以并行处理.

IPv6 地址和协议域的编码更新相对较简单,所以我们主要阐述端口域的更新.端口编码的更新包括 SRAM 和 TCAM 中的两种编码字符串的更新.

(1) 端口编码添加过程

根据新添加的端口范围跟现有端口范围的关系,我们使用图 3 中的例子(包含了 SRAM 和 TCAM 中的两种编码字符串的更新)来说明.

情况 1:新添加的端口能够找到一个层,此层上所有端口范围都不会与之重叠.例如,需要添加一个 AR 范围[8080:8081]时,可以直接在第 3 层给它分配一个/1000 的编码,同时在 TCAM 中只需添加一条相应的规则表项,不会影响到其他端口的编码及规则.

情况 2:新添加的端口在每个现有对应层都有端口跟它重叠.例如,当需要添加一个 AR 范围[5000:5010]时,由于它与[5000:5010]重叠,所以必须再添新的层,我们加入第 4 层,分配 2 比特,给[5000:5010]的编码为/010.同样地,这种更新在本方案中也不会带来其他端口编码及规则的改变.

具体操作过程是:给每个端口范围维护一个计数器.插入一个端口范围包括以下几步:¹检查这个范围是否已存在,若是,计数器加一,否则转到^o;^o若这个范围是一个 EM,则在第 2 层给它分配一个编码,若是 AR,则在第 3 层或更高层寻找一个不存在与其重叠的逻辑层,找到则»,否则动态添加一层,转到»;»在该层分配一个编码.初始化一个相应的计数器,初值为 1;并在 TCAM 中添加相应的表项.

(2) 端口编码删除过程

删除一个端口范围包括以下几步:¹在 TCAM 中删除相应表项,找到这个端口范围的位置;^o计数器减一,若值为 0,则转到»;»删除这个范围,并释放其编码.

本文给每层分配固定的比特,闲置的比特初始化为/00,更新时就只涉及包含此编码的表项的变化.通常,TCAM 的表项按照优先级从高到低存储.当加入表项的优先级比当前 TCAM 中存储的最低优先级高时需要移动规则.由于使用了 CYPRESS 10512³¹X,通过适当

的优先级编程,可以避免更新时带来其他规则表项的移动.

表 3 SRAM 存储分析

域	理论上需要的存储空间	片上 SRAM	
源/目的地址	1.152 千比特	4 个 M4K Blocks	
源端口	EM	4.608 千比特	8 个 M512 Blocks
	AR	1.12 千比特	8 个 M512 Blocks
目的端口	EM	9.216 千比特	16 个 M512 Blocks
	AR	3.36 千比特	16 个 M512 Blocks
协议号	1 千比特	2 个 M512 Blocks	
总共	20.456 千比特	50 个 M512 Blocks 4 个 M4K Blocks	41.984 千比特

注:一个 M512 Blocks 和一个 M4K Blocks 大小分别为 576 比特和 4608 比特

5 性能分析

5.1 存储需求

独立域查找需要占用的 SRAM 存储空间总共为 25.776 k bits,使用 Altera Stratix 1S80 FPGA^[15]来实现,所需的 TCAM 存储空间为 288N 比特^{XX}.

5.2 流分类查找性能

独立域的查找在 SRAM 中进行,通过流水处理,TCAM 与 SRAM 可并行操作.为了最大程度地发挥 TCAM 高速性,要求 SRAM 中独立域查找在 3 个 TCAM 周期内完成.使用 CYPRESS 10512 TCAM,时钟频率为 66MHZ,把 SRAM 配置成 198MHZ 或者更高(FPGA 片内的 SRAM 频率最高可达 319MHZ^[15]),则 SRAM 中独立域查找只需在 9 个 SRAM 周期内完成即可满足要求,见图 7 所示.

由于各个独立域编码查找是并行进行的,因此独立域查找的时间取决于其中查找时间最长的.根据第 3 节中对几种独立域编码查找的分析,最长的编码查找即 AR,最多只需要 3 个周期,完全满足独立域查找 9 个周期的时间限制.这样独立域查找最多需要 3 次 SRAM 访存周期,TCAM 需要 1 次.在流水操作时,独立域的查找可以提前进行,即一个包的独立域查找完成后,马上可以进行下一个包的独立域查找.这样在下一个包的流分类查找命令发送之前便得到了重组好的关键字,使 TCAM 的处理能力能够被完全发挥.

6 结论

本文提出了一种基于 TCAM 的高速 IPv6 流分类方

X CYPRESS 10512 Network Search Engine 的最高时钟频率为 133MHz,容量为 18Mb,按块(block)划分,可以对块进行优先级编程
XX N 为流分类表项条数

案. 该方案首先在 SRAM 中查找域编码, 然后把得到的各个域的编码组合成一个新的查找关键字, 送入 TCAM 进行联合流分类查找以得到最后的流分类结果. 采用了两种粒度的并行: 一种为独立域之间的并行查找; 另一种为独立域查找和联合流分类查找之间的并行. 分

析和仿真表明, 配置 TCAM 频率为 66MHZ, SRAM 频率为 198MHZ, 则对一个 IPv6 数据包同时进行路由查找和流分类查找可以达到 22Mpps 的处理速度, 即每 45.15ns 出一个 IPv6 流分类结果, 满足 OC192 接口线速流分类的要求.

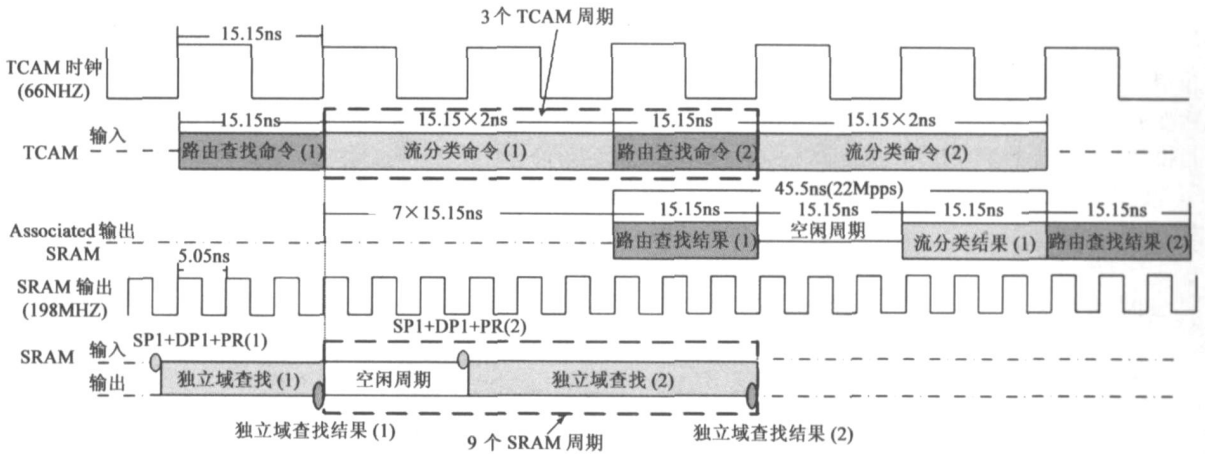


图 7 交替的路由查找和流分类时序图

参考文献:

- [1] J van Lunteren, et al. Fast and scalable packet classification [J]. IEEE Journal on Selected Areas in Communications, 2003, 21(4): 560- 571.
- [2] Nei2Fu Huang, et al. Design of mult2field IPv6 packet class2ifiers using ternary CAMs[A]. IEEE Global Telecommunica2tions Conference[C]. San Antonio: IEEE, 2001. 1877- 1881.
- [3] CYPRESS, CYNSE10512 Network Search Engine[Z]. 103- 124.
- [4] IDT, IDT75P52100 Network Search Engine[Z]. 24- 35.
- [5] E Spitznagel, D Taylor. Packet classification using extended TCAMs[A]. Network Protocols 2003. Proceedings. 11th IEEE International Conference on[C]. Atlanta: IEEE, 2003. 120- 131.
- [6] Masanori UGA, et al. A high2speed packet classification using TCAM[J]. IEICE Transactions on Communications, 2001, E82 B(9): 1877- 1881.
- [7] David Edward Taylor. Models, Algorithms, and Architecture for Scalable Packet Classification[D]. Saint Louis: Washington U2niversity in St. Louis, 2004.
- [8] Cisco, Cisco IOS Commands for the Catalyst 6500 Series Switches[Z]. 78- 83.
- [9] Internet Protocol Version 6 (IPv6) Addressing Architecture [S]. RFC 3513, 2003.
- [10] unicas2address2assignments [EB/ OL]. <http://www.iana.org/assignments/ipv6> 2005202211.
- [11] IPv6 Global Unicast Address Format[S]. RFC 3587.
- [12] class bench[EB/ OL]. [http://www.arl.wustl.edu/? det3/ClassBench/](http://www.arl.wustl.edu/?det3/ClassBench/). 200209220.
- [13] David E Taylor, et al. ClassBench: a packet classification benchmark[A]. IEEE INFOCOM 2005 [C]. Miami: IEEE, 2003. 2068- 2079.
- [14] Guansong Zhang, H Jonathan Chao. Fast packet classification using field2level trie[A]. IEEE Global Telecommunications Conference[C]. San Francisco: IEEE, 2003. 3201- 3205.
- [15] Altera, Stratix Device Handbook[Z]. 2004. 13- 16.

作者简介:

李 维 女, 1981 年生于四川, 清华大学计算机系攻读硕士学位, 主要研究方向为计算机网络, 路由查找和流分类.

刘 斌 男, 1964 年生于山东, 清华大学计算机系教授, 博士生导师. 主要研究领域为交换技术, 网络处理器, 流量工程和高速网络安全. E2mail: liub@tsinghua. edu. cn

郝 颖 女, 1977 年生于山西, 清华大学计算机系攻读工程硕士学位, 主要研究方向为信息处理和计算机网络.

林 伟 男, 1981 年生于辽宁沈阳, 清华大学计算机系攻读博士学位, 主要研究方向为流分类查找, 以及 IP 报文深度处理.

唐 毅 男, 1983 年生于湖北武汉, 清华大学计算机系攻读博士学位, 主要研究方向为流分类查找, 以及高速网络安全.