基于前缀扩展的三级索引路由查找算法

唐丽梅[1,2],邢素霞[1],陈天华[1]
(1.北京工商大学 计算机与信息工程学院,北京 100048;
2.北京英瑞博系统技术有限公司,北京 100039)

摘 要:根据路由表前缀扩展特性,采用特殊的结构构造索引表,提出了一种基于3级索引的储存表查找方法,进行流水线方式的并行查找。引入了缓冲池的思想,提出了一种改进的路由表更新方法,同时该算法支持动态更新。与基于压缩算法相比,该算法数据结构简单;与传统TCAM路由查找相比,可以节省约40%的功耗。此外,该算法在查找性能、路由更新和存储空间方面也有很大优势,能够达到最少访问一次存储器,最多需要访问3次实现处理一个IP数据包。

关键词:路由查找;前缀扩展;索引表;下一跳索引
中图分类号:TP393 文献标识码:A 文章编号:1674-7720(2012)19-0061-04

Routing lookup algorithm based on prefix expansion and 3 level index

Tang Limei[1,2], Xing Suxia[1], Chen Tianhua[1]
(1.School of Computer and Information Engineering, Beijing Technology and Business University, Beijing 100048, China;
2.Beijing Jointbest System Technology, Beijing 100039, China)

Abstract: According to the prefix expansion characteristics of routing table and the specific structure of index table, the paper puts forward a kind of stored table lookup algorithm based on 3 level index and the way of searching for assembly line parallel. It introduces the idea of buffer pool, and proposes an improved routing table update methods which at the same time supports dynamic updates. Compared with the algorithm based on the compression, this algorithm is simple on data structure. Compared with the traditional TCAM routing search, it can save about 40% power consumption. In addition, there are great advantages in search performance, routing update and storage space aspects. To deal with an IP packets it only needs to visit a memory at least once, and three times at most.

Key words: routing lookup; prefix expansion; index table; NHI

随着网络的高速发展,Internet的网络传输量每几个月就增加一倍,这也给高速路由的设计带来了挑战,骨干网路由器接口速率已经达到Tb/s量级,IP路由查找操作已经成为路由器转发性能乃至Internet整体性能的主要瓶颈之一。因此提高路由查找速度的关键是采用快速的路由查找算法。

路由器IP路由查找面临巨大挑战,主要表现在:(1)实现最长前缀匹配的路由查找算法设计困难[1];(2)路由表庞大,查找记录条目极多;(3)路由更新频繁,最高每秒更新条目达几万条;(4)接口速率越来越高,OC-768(40 Gb/s)以太网及更高标准要求。实现100 Gb/s接口的线速转发,包转发率要达到150 Mb/s,每包处理时延小于6.72 ns。

快速的路由查找技术一直是一个热门课题,近年来提出了不少路由查找算法,传统的基于软件的路由查找算法已经不能满足分组的线速转发,目前高性能核心路由器基本上都采用基于硬件的路由查找算法。路由查表实现的主要功能就是最长前缀匹配(Longest Prefix Matching)。基于前缀值的二分搜索、页面压缩等基于树的搜索存储空间占用少,利用率高,但由于算法实现复杂,硬件实现上不合适。TCAM(Content Addressable Memory)[2]采用保存关键字掩码的方式来保存任意长度的关键字表项,并且使用并行比较,仅在一个时钟周期内就可以完成一次查表操作,能够实现高速查表。但是

TCAM 的路由表更新操作复杂、功耗大[6]、容量小且价格昂贵。这些缺点使研究人员考虑用基于 SRAM 的算法来实现 LPM。

在基于 SRAM 的算法中，SRAM 每比特存储需要 6 个晶体管，功耗低，TCAM 每比特的价格是 SRAM 的 30 倍。可见，一个好的基于 SRAM 的算法比 TCAM 更具吸引力。由于路由查表的速度和复杂性的需要，采用单一的查找算法达不到理想的速度和效率，因此应采用多种算法的综合以及辅助策略。

本文介绍的路由查找算法利用前缀扩展的特性，构造三级索引表，并利用流水线并行方式查表，最少一次访问存储器，最多 3 次访问存储器就能查找到包转发信息(输出端口、下一跳 IP 地址等)。由于对数据结构作特定优化，能支持动态分配表项空间。对更新算法加入缓存的思想，大大减小了地址占用和申请的开销。

1 查找算法原理

1.1 前缀扩展特性[3]

本算法主要利用前缀扩展的特性，采用特定的结构来构造索引表。前缀扩展是将长度不同的前缀集合中所有小于最长前缀长度的前缀统一扩展为不小于最长前缀长度的集合。例如，有前缀集合 P={0*，10*，111*，11001*}，其中最长前缀长度为 5，把所有长度小于 5 的前缀加以扩展，扩展结果如表 1 所示。经扩展后，集合 P 形成一个新的集合 P={00000*，00001*，…，10111*，11100*，11101*，11110*，11111*}。前缀扩展的目的是把任意长度的前缀变成固定长度的前缀来简化查找操作。

表 1 前缀扩展方法

P 前缀集合元素	扩展后元素
0*	00000*，00001*，00010*…，01111*；
10*	10000*，10001*，10010*…，10111*；
111*	11100*，11101*，11110*…，11111*；
11001*	11001*

注：* 表示 IP 地址除前缀外的 bit。

1.2 算法实现

三级索引[4]的具体结构如图 1 所示，其中给出如下定义：

需要扩展的前缀为 Prefix=P（如 P=111000*），前缀长度为 Prefix length=PL，该前缀对应的下一跳地址为 next hop=NH，端口号 port number=No。

根据骨干网路由表前缀长度分布[1]，可以发现，前缀长度几乎分布在 8~32，而且分布极不均匀，长度在 16~32 的前缀约占总数的 99%；前缀长度为 24 的路由最多，约占总数的 45%。可见路由前缀小于 8 和大于 24 的表项非常少，因此小于 8 或大于 24 的前缀长度扩展的前缀集合绝大多数索引集为空，造成索引表的利用率非常低，而且随索引前缀长度增加以指数衰减。根据这一特性，在本算法中做划分前缀改进。通过扩展存储长度小于 16 的前缀构造一级索引表；通过扩展存储长度不

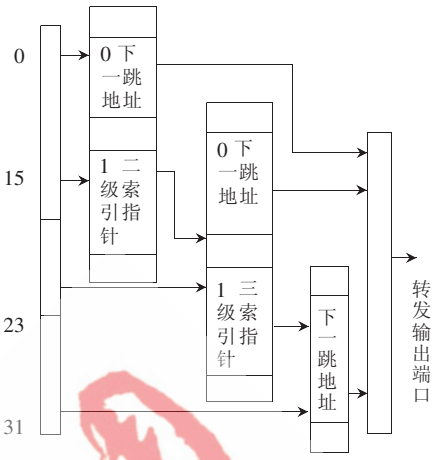

图 1 三级索引查表算法结构图

大于 24 的前缀二级索引表；对于长度大于 24 的前缀不进行扩展，由于其数量非常少，进行前缀扩展就显得浪费空间。

本查找算法中对第 3 级索引表的查找方法进行改进。在二级索引表项中设置偏移量标志 Rel。根据所有前缀中最大前缀长度为 L_m，设 IP 地址的前 24 bit 相同的所有前缀中最大前缀长度为 L_m，定义偏移量 Rel 为：

$Rel=L_m-24$；

根据 Rel 的值动态分配一个独立的 SRAM 用于储存三级索引表项，最大需要地址空间为 2^{Rel}。这样，整个路由表的大小将被控制在一个较小的规模，而不用引入复杂的位图压缩[5-6]来管理储存空间。

动态分配二级表的存储空间，根据一级表项中标志位 1 分配一个连续的 256 个 SRAM 地址空间，并将首地址作为基地址存储在一级表中。对三级表分配独立的 SRAM。因此分别对 3 个表进行流水线查找，同时进行查找操作，使查表速度迅速提升。

第一级索引表地址从 0~32 767，一共有 32k 个(2^{15})地址空间，每个地址空间存储的数据结构有两种：下一跳信息或指向二级表的指针，$Rel=24-L_m$(前 16 bit 中最大前缀长度)。其表项结构分别如图 2 和图 3 所示。

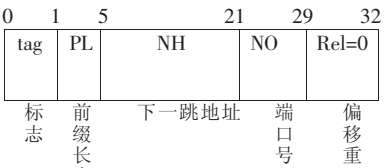

图 2 一级索引表中用于存储下一跳信息的表项结构

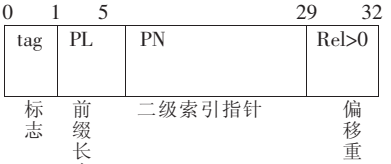

图 3 一级表中用于存储二级表指针信息的表项结构

每个一级表表项共有 32 bit，第 0 bit 是指示标记(tag)，用于指示该表项存储的是路由信息还是指针信

息，即当 tag=0 时表示该表项存储的是下一跳路由信息（简称路由表项），包括路由前缀长度（PL）、下一跳 IP 地址（NH）、端口号（No）以及偏移量信息，如图 2 所示；tag=1 时表示该表项存储的是指向二级索引表的地址索引信息，如图 3 所示。当 tag=0 时，第 1~4 bit 表示路由前缀长度，5~20 bit 用于标记下一跳 IP 地址，21~28 表示输出索引信息，29~32 表示输出端口号。考虑到长度在 16~24 bit 之间的前缀占 99.93%，设置第二级索引表，扩展目的 IP 的第 3 字节，每个二级表从 0 到 255，一共有 256 个地址空间。每个地址空间存储的数据结构也有两种（和一级表类似）：下一跳信息或指向二级表的指针，其表项结构与一级索引结构类似。

对于长度小于 16 的路由前缀，根据一级索引表中的下一跳 IP 地址即可完成查找。对于长度大于 16 小于或等于 24 的路由前缀，需要同时对一级表和二级表进行存储，首先在其前 16 bit 对应的一级表地址空间中存储二级表的指针信息，而剩下的比特位则同样通过前缀扩展的方法扩展到 8 bit，然后再存储到二级表中即可，扩展后在相应的表项中存储的都是下一跳路由信息。二级表的表项结构如图 4 所示。

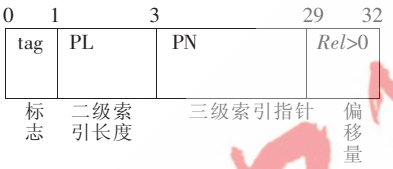

图 4 二级表中用于存储三级表指针信息的表项结构

对于长度大于 24 的路由前缀，需要同时对一级表、二级表和三级表进行存储，首先在其前 16 bit 对应的一级表地址空间中存储二级表的指针信息，而中间 8 bit 对应的二级表地址空间中存储的是三级表的指针信息，对于剩下的比特位，根据其长度，根据二级索引表指针可在三级表的动态存储空间中快速找到第三级地址，然后将路由信息存入其中，三级表中表项信息获取完成后，查表过程结束。

2 路由表的更新

本算法的更新过程与查找过程都是先根据前缀对应的分段和索引查找到对应的子表，然后在其涉及的范围内读取各个表项，再根据表项的值确定是否用新的路由前缀信息覆盖该表项。如果在查找该段前缀时，该表没有相应的段空间，则需在储存模块中分配相应的存储单元，当某段地址空间为空时，收回该储存空间。路由表需要更新的时候，首先 CPU 根据前缀的长度进行扩展，送入 FPGA 进行判断，根据路由表项信息完成插入和删除操作。

2.1 一级索引表更新

对于长度不大于 16 的前缀 L_i，首先进行前缀扩展，得到 m 个扩展前缀，在一级表中读出以 $L_i(i=1,2,\cdots,m)$ 为地址的内容，若该内容是路由信息或是空白信息，则不作处理，将新的路由信息写入一级表中的地址中即可；若该内容为二级表的指针信息，那么将该指针信息作为二、三路由表对应的 SSRAM[7]中要释放的地址块号送到一级索引的 SSRAM 存储空间管理模块中。插入过程如图 5 所示。

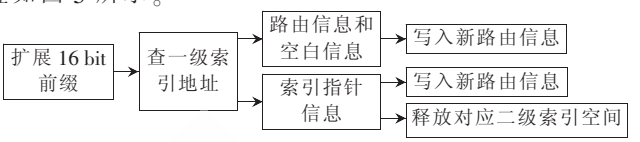

图 5 一级索引表更新

2.2 二、三级索引表更新

对于大于 16 的前缀长度的二、三级表更新算法，首先扩展到 24 bit 前缀。更新方式与一级索引表更新类似。此外，本算法引入 Freeunit[2]来优化空间管理，改进更新操作。每种长度的地址前缀块分配一个 Freeunit，其容量根据前缀长度而定。较长的前缀块对应的分配容量较大，反之，容量较小。Freeunit 的实现可以采用堆栈或队列等数据结构来实现。二、三级索引表更新前缀项进出缓存池过程如图 6 所示。

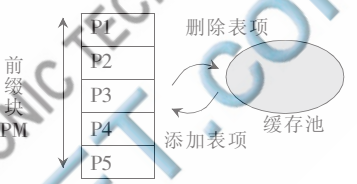

图 6 前缀项进出缓存池过程

把 Freeunit 作为一个缓冲池 BP，当删除一个表项时，只是单独的将表项内容删除，而不改变前缀块和整个三级索引表项结构，同时将删除的表项放入到 Freeunit 中；当添加新的表项时，从对应的 Freeunit 中取出当前的空闲表项，直接添加，由此可以大大减少因为表项的添加和删除而释放地址空间所需时间，申请新的地址空间造成的时间开销。

硬件结构设计如图 7 所示，三级索引结构包括 3 个流水线并行查找模块和 3 个更新单元。每个表都需要一个单独的 SSRAM 进行存储，这样，一共就使用 6 块 SSRAM。对于一级索引，有 30 720 个表项，每个表项有 64 bit（8 B）存储空间，则一个一级表就需要 237.5 KB 大小的 SSRAM，所以每个一级表选用了一个 500 KB 容量的 SSRAM。第二级索引表和第三索引表的大小不能精确确定，选用了一个 4 MB 容量的 SSRAM 和 2 MB 容

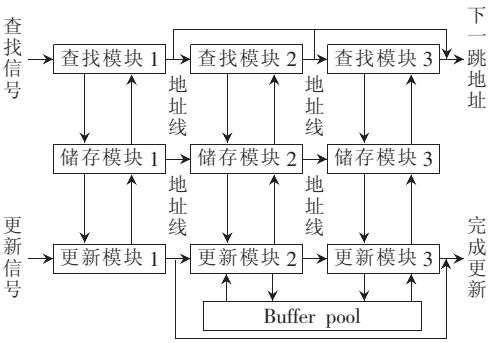

图 7 三级流水线并行处理算法硬件结构

表 2 算法查找速度比较表

查找速度(Mp/s*) / 查找方法	路由表项数					
	100	200	500	1 000	10 000	100 000
动态前缀树查找	1.000(7)	0.909(8)	0. 833(9)	0.769(10)	0. 500(13)	0.345(16)
二进制 trie 树	1.000(9)	0.900(9)	0.810(10)	0.623(14)	0.471(16)	0.201(20)
二级索引页面压缩算法[参考文献 3]	12.518(1)	12.033(1)	11.614(1.2)	8.065(1.4)	6.662(2.0)	6.110(2.3)
四级流水线 RAM 查找算法[参考文献 1]	95.434(1.8)	93.013(1.8)	89.622(1.9)	88.071(1.9)	70. 821(2.0)	40.524(2.0)
三级索引 SRAM 快速查找(本文所用算法)	105.647(1)	100.044(1)	98.189(1.2)	94.011(1.4)	88.829(2.0)	50.641(2.0)

注: Mp/s= Million packets per second;括号中的数据表示查找平均访问存储器次数。

量的 SSRAM。而对于第三级表,由于其容量很小,因此将三级表和二级表放在同一块 SSRAM 里。每个查找模块都可从输入端或寄存器中读取表项信息,解析 IP 地址位[8],访问存储器获取数据,最后将数据写入寄存器或者送到输出端进行输出转发操作。

3 实验结果及算法比较

实验环境为:Celeron,500 MHz Windows XP,256 MB RAM。索引算法用 FPGA 实现,采用 Xilinx 公司的 spartan-6 系列芯片。它们可以提供丰富的片内 SRAM 资源,均以 SRAM 为储存介质。为了测量性能,通过随机数的方法产生随机 IP 地址、掩码和端口索引号,用数组记录这些信息,在添加路由表前记录系统时钟,添加完成后又记录一次系统时钟,进行大量的插入操作后计算完成一次操作所用的时间。查找与删除操作也采用同样的方法来测量。通过对比可知,三级前缀划分并改进第 3 级索引可有效提高地址空间利用率,减少空索引集数量,加入缓存的更新算法有效减少了更新开销时间,从而提升查找速度。由于查找需要一个时钟周期,而时钟频率为 100 MHz,因此每秒可以完成 100 M 次查找,假设报文长度均为 40 B,可以满足 20 Gb/s 的链路速率。

算法查找速度比较表如表 2 所示,算法存储容量比较表如表 3 所示。从实验结果来看,当表项数目较小时,二进制 trie 树查找[9]过程表项次数较多,相应的查找速度也较慢。随着表项数目的增加,查找速度变化非常缓慢,已经不能适用于快速的路由查找。对于动态前缀树查找方法,查找中表项比较的次数随表项数目变化的速度比较缓慢,相应的查找性能变化比较平缓,基本维持在同一个数量级上,平均查找速度低。二级索引页面压缩算法查找速度随查找表项的数目变化的速度较缓慢,相应的查找性能较好,由于压缩处理,表项占用空间很小。缺点是压缩算法[10]用硬件实现不合适。四级流水线查找速度快,访存次数少,此算法使用的存储容量比较大,特别是在表项数目较多的情况下。与三级索引 SRAM 快速查找 RAM 的查找算法相比,三级索引算法具有很快的查找速度,甚至当表项数目达到 100 000 时,仍然可以达到 50 Mp/s 多的查找速度。从存储容量上来看,较四级流水线 RAM 查找存储容量更小。由比较可知,三级索引 SRAM 快速查找在查找速度、储存空间、更新速度方面都具有优势。该算法非常适于需要高速报文转发的网络环境。

表 3 算法存储容量比较表

查找速度(Mp/s*) / 查找方法	路由表项数					
	100	200	500	1 000	10 000	100 000
动态前缀树查找	2.1	4.2	10.5	20.9	207	1 546
二进制 trie 树	5.8	10.9	15.7	23.4	232	1 996
二级索引页面压缩算法[参考文献 3]	1.9	4.5	11.7	22.5	239	1 677
四级流水线 RAM 查找算法[参考文献 1]	144	158	204	279	7 953	14 572
三级索引 SRAM 查找(本文所用算法)	90	118	133	165	3 320	4 500

本文提出基于三级索引来实现路由查表算法,并利用前缀扩展和引入更新缓存的技术来实现优化。最快的查找只要一次访问存储器就可以找到端口索引,获得下一跳信息需要 2 次访问存储器。最多 3 次访问存储器就可以获得端口索引。在实现高速查找的同时,兼顾到存储空间利用率和实现复杂度等多种因素,比单纯使用四级流水线查找速度提高了 15%;对第 3 级索引表采用动态管理,节省了 30%储存空间。

参考文献

[1] 王波. 基于 FPGA 的快速路由查找算法研究及实现[D]. 西安:西安电子科技大学,2009.

[2] 苗建松,丁炜. 改进的 TCAM 路由更新方法与实现[J]. 微电子学与计算机,2006,23(10):144-149.

[3] 刘亚林,刘东.基于前缀扩展的快速路由查找算法[J].计算机学报,2001,24(12):1272-1278.

[4] 张毅,郭玲丽.基于 FPGA 的高速路由查找算法[J].电子元器件应用,2009,11(9):22-27.

[5] 彭元喜,唐玉华,龚正虎.基于压缩 NH 表的高速 IP 路由查找算法的研究[J].电子学报,2002,(2):32-36.

[6] 王利媛,马跃,徐塞虹.对路由表结构和查找算法的研究[J].计算机应用,2004(11):10-12.

[7] MCEWAN A A, SAUL J. A high speed reconfigurable firewall based on parameterizable FPGA-based content addressable memories[J]. The Journal of Supercomputing, 2001,19(1):93-103.

[8] IOANNIDIS I, GRAMA A, ATALLAH M J. A-daptive data structures for IP lookups [J]. AlM Journal of Experimental Algorithms, 2005(10):75-84.

[9] 谭兴晔,张勇,雷振明.基于快速搜索树的路由查表算法[J].计算机应用研究,2005(7):231-235.

[10] 徐恪,吴建平,吴剑.路由查找算法评价系统的设计与实现[J].小型微型计算机系统,2003,24(2):274-276.

(收稿日期:2012-04-07)

作者简介:

唐丽梅,女,1988 年生,硕士研究生,主要研究方向:计算机通信、FPGA 储存、磁盘阵列。