基于压缩 NH 表的高速 IP 路由查找算法的研究

彭元喜,唐玉华,龚正虎
(国防科学技术大学计算机学院,湖南长沙 410073)

摘 要: 由于因特网速度不断提高、网络流量不断增加和路由表规模不断扩大,IP 路由查找已经成为制约核心路由器性能的主要原因,因而受到了广泛重视. 目前人们已经提出几种高速 IP 路由查找算法,但没有一种是理想的. 本文提出一种使用压缩 NH 表进行 IP 路由查找的方法,它具有查找速率高、更新时间快、存储代价低、易于实现等特点,能满足 10Gbps 速率核心路由器环境的要求.

关键词: 下一跳表; IP 路由查找; 核心路由器
中图分类号: TP393 **文献标识码**: A **文章编号**: 0372-2112 (2002) 02-0196-04

The Study on High-Speed Algorithms of IP Routing Lookups Based on the Compressed Next Hop Table

PENG Yuan-xi, TANG Yu-hua, GONG Zheng-hu
(Computer School, National University of Defense Technology, Changsha, Hunan 410073, China)

Abstract: The problem of IP routing lookups has received much attention recently because it is one of the main reasons why performance of core routers in the Internet becomes the bottleneck. Several algorithms for solving this problem have been proposed, but none is ideal. In this paper we propose an algorithm of IP routing lookups using compressed next hop table. The algorithm has characteristics such as fast search time, fast update time, small memory space and easy implementation, and can be used in core routers that have 10Gbps interfaces.

Key words: NH(Next Hop) table; IP routing lookup; core router

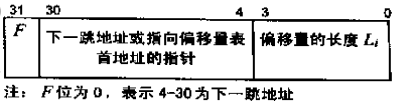
1 引言

由于因特网速度不断提高、网络流量不断增加和路由表规模不断扩大,核心路由器已经成为制约因特网性能的主要瓶颈,其主要原因在于核心路由器需要执行复杂的 IP 路由查找操作. 目前,无论在学术界还是在工业界,IP 路由查找问题均已受到广泛重视.

最近,人们已经提出许多机制和算法解决 IP 路由查找问题[2~6,11~15],但没有一种是理想的. 这些机制和算法可分为软件[3~6,11]和硬件[2,12~15]解决方法,本文在已有软硬件解决方法基础之上提出一种使用压缩 NH 表进行 IP 路由查找的方法,它使用可变长偏移量表并对 NH 表用图论压缩方法进行压缩. 该算法的最大特点是所需存储代价低,能够满足核心路由器 10Gpbs 端口速率的要求,能够减少路由器成本和硬件设计复杂度.

2 提议的机制

实现查找机制最直接的方法是为每个 32 位 IP 地址设计一个表项,这样,转发表的大小为 4G. 为减少转发表大小,可以使用压缩 NH 表进行 IP 路由查找,它使用可变长偏移量表并对 NH 表用图论压缩方法进行压缩. 该方法共使用三张表:段表、偏移量表和压缩 NH 表. 段表大小为 64k,地址从 0.0 到 255.255,存储所有长度小于或等于 16 的前缀的 NH 地址;偏移量表存储位向量(Bit Vector,BV),当前缀长度大于 16 位时,用于计算在压缩 NH 表中查找 NH 地址的索引,偏移量的长度为该段中所有前缀最长长度减去 16,因此每一段的偏移量长度均不一样,以 δ_i ($0 \le i < 64k$) 表示之,并存储在段表表项中;压缩 NH 表存储压缩以后的 NH 地址. 段表表项的格式如图 1.



注: F 位为 0,表示 4-30 为下一跳地址
为 1,表示 4-30 为指向偏移量表首地址的指针

图 1 段表表项格式

偏移量表可以直接存储 NH 表,其每一表项均存储一个 NH 地址,由于大部分表项为前缀的扩展,许多表项的内容是相同的,可以对这些重复的表项进行压缩,将连续重复的表项压缩为一个表项,这样可大大减少转发表的大小. 考虑到偏移量表项的内容可以通过图的边上的权值来表示,因此我们用图论建立一种压缩方法. 基本思想说明如下:将偏移量表用路

收稿日期:2000-05-10;修回日期:2001-10-12
基金项目:国家 863 重点攻关项目"核心路由器"(863-300-01-03-99).

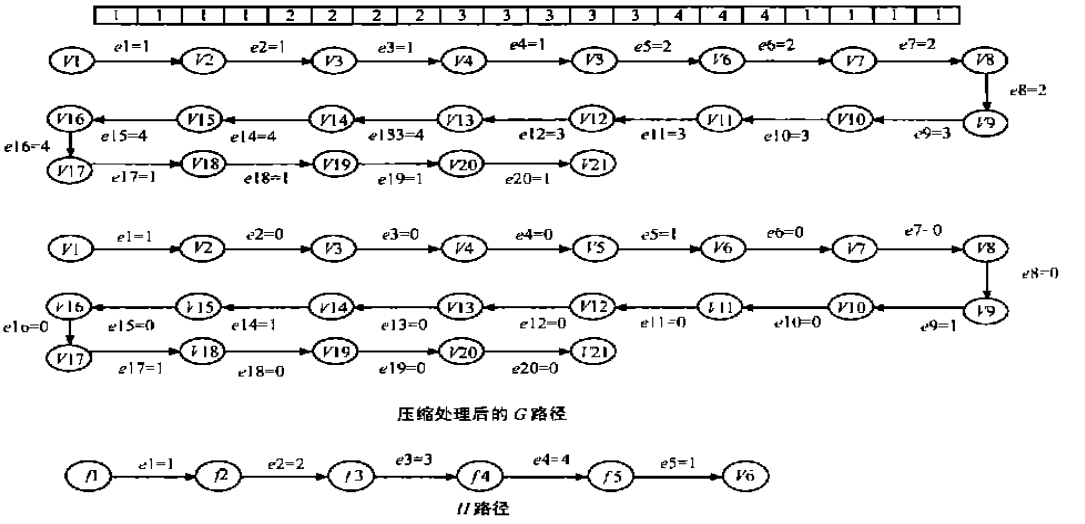


图 2　用图论建立的一种压缩方法

径 G 表示,压缩后的 NH 表用路径 H 表示,请参见图 2.

偏移量表的所有表项表示成路径 G 的边,设偏移量表的表项数为 $k1 = 2^{\delta}$,则路径 G 的边数也为 $k1$,记为 $e_1, e_2, \cdots, e_{k1}$,路径 G 的顶点数为 $k1 + 1$,记为 $v_1, v_2, \cdots, v_{k1+1}$,设边的权值记为 $c_i (i = 1, 2, \cdots, k1)$,记录偏移量表相应表项的 NH 地址,顶点的权值记为 $w_i (i = 1, 2, \cdots, k1 + 1)$,初始值全为 0. 由于每一个前缀只有一对起点和终点,m 个前缀的起点和终点总数最大为 $2m$,$2m$ 个点所产生的非重叠区间不会大于 $k2 = 2m - 1$,所产生的压缩 NH 表的表项数不会大于 $k2$,故设路径 H 的边数也为 $k2$,记为 $f_1, f_2, \cdots, f_{k2}$,路径 H 的顶点数为 $k2 + 1$,记为 $u_1, u_2, \cdots, u_{k2+1}$,设边的权值记为 $d_i (i = 1, 2, \cdots, k2)$,记录压缩 NH 表项的 NH 地址,初始值全为 0. 开始时 $k3 = 1$, $d_{k3} = c_1$, $w_2 = c_1$, $l_2 = c_1$, $c_1 = 1$,起点 $q = v_3$, $k4 = 1$

(1) 对于起点 q 及其后续节点 v_i, $w_i = w_{i-1} + c_{i-1}$, $k_4 = K4 + 1$, $l_i = w_i / K4$

(2) 如果 $l_i = l_{i-1}$, $c_{i-1} = 0$;

(3) 如果 $l_i \neq l_{i-1}$, $k3 = k3 + 1$, $d_{k3} = c_{i-1}$, $w_i = c_{i-1}$, $l_i = c_{i-1}$, $c_{i-1} = 1$,起点 $q = v_{i+1}$, $k4 = 1$

重复 1,2,3 步,直至到达路径尾.

对于压缩处理后的 G 路径,其每条边上的权值只有 0 和 1 两种状态,可用 1 位表示,因此其所有边可以用一个 2^{δ} 长的位向量(Bit Vector,BV)表示. 每个分组值对应于位向量中的某一位,搜索时只需计算该位前面 1 的个数,即可求出相应 H 路径边对应的 NH 地址.

2.1　NH表的压缩算法

压缩算法完成由一组前缀到相应的 BV 与压缩 NH 表的变换,它基于如下观察:对于任何两个前缀,它们所包含的范围要么不相交,要么一个完全包含另一个,即不存在部分重叠的情况.

设 $P = \{p_0, p_1, \cdots, p_{m-1}\}$ 为给定段的一组按前缀长度排序的前缀集合,用 l_i 和 d_i 分别表示前缀 p_i 的长度和下一跳地址,对于集合中任何一对前缀 p_i 和 p_j, $i < j$ 的充要条件为 $l_i < l_j$;令 $K_i = l_i - 16$;该段的偏移量长度 δ(为表示方便,我们以 δ 代替上面的 δ_i)为 $l_{m-1} - 16$;用 S_i 和 E_i 分别表示前缀 p_i 的起点与终点,$p_i = a.b.c.d$, $c_0, c_1, \cdots, c_{15}$ 为 $c.d$ 的二进制形式,$a_0, a_1, \cdots, a_{\delta-1}$ 为起点的屏蔽,其中 $j < K_i$ 时 $a_j = 1$, $j > K_i$ 时 $a_j = 0$, $b_0, b_1, \cdots, b_{\delta-1}$ 为终点的屏蔽,其中 $j < K_i$ 时 $b_j = 0$, $j > K_i$ 时 $b_j = 1$,则 $S_i = (c_0, c_1, \cdots, c_{\delta-1} \& a_0, a_1, \cdots, a_{\delta-1})$, $E_i = (c_0, c_1, \cdots, c_{\delta-1} | b_0, b_1, \cdots, b_{\delta-1})$. 这意味着对于 $S_i < j < E_i$, $NH_j = d_i$;同时也意味着前缀等价于数轴上的区间 $[S_i, E_i]$. 由于前缀之间的完全包含关系,区间之间也是完全包含的关系. 根据最长前缀匹配技术的要求,在被包含区间之内应存储对应于被包含区间的长度较长的前缀的 NH 地址. 根据上述图论压缩方法以及区间之间的完全包含关系,下面我们对不同的 m(分段后的前缀数)取值建立两种压缩算法.

压缩算法 1:

(1) $BV = 0$, $CT = 0$.

(2) 对于给定段的一组按前缀长度排序的前缀集合 $P = \{p_0, p_1, \cdots, p_{m-1}\}$,计算对应前缀的起点和终点,并构成集合数组 $T = \{S_0, E_0, S_1, E_1, \cdots, S_{m-1}, E_{m-1}\}$,用 $\{Addr, NH, F\}$ 记 T 中的元素,其中 $Addr$ 表示元素在数组上的值,NH 表示元素 NH 地址,F 为标志位,对于起点元素,$F = S$,对于终点元素,$F = E$.

(3) 将 T 中各前缀的起点和终点按数轴上的顺序排序,对于相等的两点保持顺序不变. 对于集合数组 T 中任何一对元素 $T[i]$ 和 $T[j]$, $i < j$ 的充要条件为 $T[i].Addr < T[j].addr$

(4) 对于 T 中的终点元素,表示一个区间结束,该区间将包含区间分成两个不同的部分,这时需增加第二部分的起点,该起点在数轴上的位置为 $Ei + 1$. 新起点元素的计算方法如下:

```
for i = 0 to 2m - 1{
  if (T[i].F = = E){ j1 = i - 1; j2 = 1;
    while (j2 > 0){
```

```
if( T[j1]. F = = S){ j2 - - ; j1 - - ;}
      else{ j2 + + ; j1 - - ;}   }
if( j1 ≧0)    T[ i ]. NH = T[j1]. NH;
   else       T[ i ]. NH = Null ;
 T[ i ]. Addr + + ;}}
```

(5) 对于 T 中的任何元素 $T[i].Addr$ > 前缀 $p0$ 的终点，删除它

(6) for $i = 0$ to $2m - 1$ { $BV_{T[i].Addr} = 1$; $CT_i = T[i].Addr$; }

压缩算法 1 与 2 的讨论：

A. 正确性：算法 1 的正确性证明（仅证明 $m = 3$ 的情况，$m > 3$ 的情况与此类似）：设 $P = \{p_0, p_1, p_2\}$ 为给定段的一组按前缀长度排序的前缀集合，假定前缀 $p0$，$p1$，$p2$ 的起点、终点、长度分别为 $S0$、$E0$、$L0$，$S1$、$E1$、$L1$，$S2$、$E2$、$L2$，则有 $L0 \leq L1 \leq L2$.

(1) 若 $p0 \supseteq p1 \supseteq p2$，由于区间之间的完全包含关系，则有 $S0 \leq S1 \leq S2 \leq E2 \leq E1 \leq E0$.

①$E0 = E1 = E2$，则 $T = \{S0, S1, S2, E0, E1, E2\}$，根据步骤 4，$E0$、$E1$、$E2$ 的 $Addr$ 加 1，其 NH 值分别为 $S1$ 的 NH 值、$S0$ 的 NH 值、Null，根据步骤 5，$E0$、$E1$、$E2$ 将被删除；②$E0 > E1 = E2$，则 $T = \{S0, S1, S2, E1, E2, E0\}$，根据步骤 4，$E0$、$E1$、$E2$ 的 $Addr$ 加 1，其 NH 值分别为 Null、$S1$ 的 NH 值、$S0$ 的 NH 值，根据步骤 5，$E0$ 将被删除；③$E0 = E1 > E2$，则 $T = \{S0, S1, S2, E0, E1, E2\}$，根据步骤 4，$E0$、$E1$、$E2$ 的 $Addr$ 加 1，其 NH 值分别为 $S0$ 的 NH 值、Null、$S1$ 的 NH 值，根据步骤 5，$E0$、$E2$ 将被删除；④$E0 > E1 > E2$，则 $T = \{S0, S1, S2, E2, E1, E0\}$，根据步骤 4，$E0$、$E1$、$E2$ 的 $Addr$ 加 1，其 NH 值分别为 Null、$S0$ 的 NH 值、$S1$ 的 NH 值，根据步骤 5，$E0$ 将被删除；因此，$p0 \supseteq p1$，$p0 \supseteq p2$ 时，正确性成立.

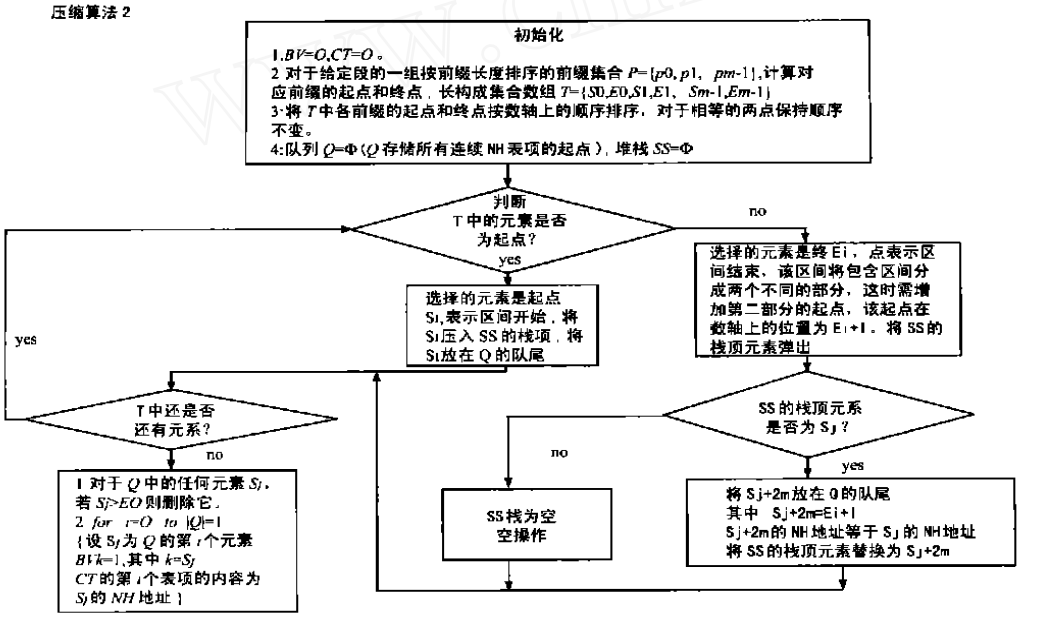
(2) 若 $p0 \supseteq p1$，$p0 \supseteq p2$，$p1$ 和 $p2$ 不相交，设 $p1$ 的范围在 $p2$ 之前，$p1$ 在 $p2$ 之后的证明与此相同，由于区间之间的完全包含关系，则有 $S0 \leq S1 \leq E1 \leq S2 \leq E2 \leq E0$.

(a) $E1 = S2$ 的情况是不存在的，因为 $p1$ 和 $p2$ 不相交.

(b) $E1 < S2$ 的情况

①$S0 \leq S1 \leq E1 < S2 \leq E2 < E0$，则 $T = \{S0, S1, E1, S2, E2, E0\}$，根据步骤 4，$E0$、$E1$、$E2$ 的 $Addr$ 加 1，其 NH 值分别为 Null、$S0$ 的 NH 值、$E1$（即 $S0$）的 NH 值，根据步骤 5，$E0$ 将被删除；②$S0 \leq S1 \leq E1 < S2 = E2 = E0$，则 $T = \{S0, S1, E1, E0, S2, E2\}$，根据步骤 4，$E0$、$E1$、$E2$ 的 $Addr$ 加 1，其 NH 值分别为 Null、$S0$ 的 NH 值、$S0$ 的 NH 值即 Null，根据步骤 5，$E0$、$E2$ 将被删除；③$S0 \leq S1 \leq E1 < S2 < E2 = E0$，则 $T = \{S0, S1, E1, S2, E0, E2\}$，根据步骤 4，$E0$、$E1$、$E2$ 的 $Addr$ 加 1，其 NH 值分别为 $E1$（即 $S0$）的 NH 值、$S0$ 的 NH 值、Null，根据步骤 5，$E0$、$E2$ 将被删除；因此，$p0 \supseteq p1$，$p0 \supseteq p2$，$p1$ 和 $p2$ 不相交，$p0 \supseteq p1$，$p0 \supseteq p2$ 时，正确性成立. 证毕.

压缩算法 2



压缩算法 2 的正确性证明与压缩算法 1 的正确性证明类似（略）.

B. 复杂性：从压缩算法 1 的描述过程可以看出，算法过程只需用结构与数组来实现，实现过程非常直观、简洁，总的时间复杂性为 $O(m^2)$，当 m 较小时，是一种比较实用算法. 由于分段之后 m 的值一般较小，因此压缩算法 1 具有实用价值. 但是，也可能出现分段后 m 较大的情形，此时我们考虑用压缩算法 2 代替压缩算法 1，不难得到压缩算法 2 的时间复杂性为 $O(m)$，相对于压缩算法 1 而言，由于压缩算法 2 采用队列与堆栈，因此实现过程相对复杂. 在实际应用中，可以根据 m 的大小选取相应算法.

C. 最优性：由于每一个前缀只有一对起点和终点，m 个前缀的起点和终点总数最大为 $2m$，因此压缩算法 2 从时间复杂性的角度来讲为最优算法.

2.2 搜索

为便于搜索，在偏移量表中存储 BV 时，将 BV 分成为 16

位长的 BVA 序列,同时对于每个 BVA 均存储一个 16 位长的 BVB,BVB 的值等于先前所有 BVA 中 1 的累加和. BVB 的值也可以递归计算:第 0 个 BVB 的值为 0,第 $i(i \geq 1)$ 个 BVB 的值等于第 $i-1$ 个 BVB 的值加上第 $i-1$ 个 BVA 中 1 的个数. 第 i 个 BVA 和第 i 个 BVB 构成偏移量表的第 i 个表项.

搜索时将 IP 地址分成两部分:段(高 16 位)和偏移量(可变长,为该段中所有前缀的最长长度减去 16),段用于查找段表,偏移量用于查找偏移量表. 设偏移量值为 x,则 x 对应的偏移量表项为 $y=(x \text{ DIV } 16)$,令 $a=(x \text{ MOD } 16)$,则 a 表示 x 在第 y 个表项的 BVA 中所对应的位,设 A 表示该 BVA 第 0 位至第 a 位中 1 的个数. 设 z 为第 y 个表项的 BVB 与 A 之和,则偏移量 x 所对应的 NH 地址为压缩 NH 表的第 $z-1$ 个表项的内容.

2.3 硬件实现

本机制的硬件实现示意图见图 3.

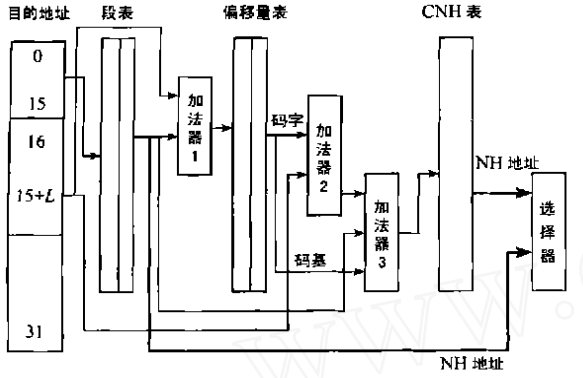


图 3　硬件实现示意图

当 IP 分组到达时,首先用其目的地址的高 16 位(0～15 位)作为索引查找段表,段表相应表项的第 31 位决定第 30～4 位的内容是下一跳地址,还是指向偏移量表首地址的指针,若为指向偏移表首地址的指针,表示需查找偏移量表,同时在第 3 至 0 位给出偏移量长度 δ. 查找偏移量表时用目的地址的第 16～$(\delta+15)$ 位作为偏移量,对应的偏移量表项为第 16～$(\delta+11)$ 个表项,查找的结果为相应的 BVA 和 BVB,因为只有 BVA 前面的 x 位需要计算 1 的个数,这里 x 为目的地址的第 $(\delta+12)$～$(\delta+15)$ 位之值加 1,因此可将后面的 $(16-x)$ 位屏蔽掉,然后用加法器 2 计算 BVA 中 1 的个数,其结果与 BVB 和指针相加,得到查找压缩 NH 表的索引,从而得到最后结果.

3 性能分析

本文使用 IPMA 工程[7]所提供的路由表评价其性能,对路由表按路由表项前缀的高 16 位进行分段分析,高 16 位前缀相同的路由表项属于相同的段,并统计段内部所有前缀中最长前缀长度为 i 的段数 $x[i]$ $(i=1\cdots16)$,则本算法所需存储器大小的计算公式为:

$$M_{cnh}=M_s+M_o+M_c \tag{1}$$

其中:

$$M_s=2^{16}*4 \text{ (Bytes)} \tag{2}$$

$$M_o=\left(\sum_{i=4}^{16}x[i]*2^{i-4}*4\right)+\left(\sum_{i=1}^{3}x[i]*4\right) \text{ (Bytes)} \tag{3}$$

$$M_c=\sum_{j=1}^{2^{16}}r[j]*2*4 \text{ (Bytes)} \tag{4}$$

其中 4 表示一个表项占用 4 字节,$r[j]$表示第 j 段所包含的前缀数,2 表示一个前缀在压缩 NH 表中占用两个表项:起点和终点.

表 1 为本算法与文献[2]所提出的 DIR-24-8 和 DIR-21-3-8 机制和文献[13]所提出的 RAM 算法的性能比较结果.

表 1　各种查找机制比较

查找机制	访存次数(最小/最大)	所需存储器大小
本机制	1/3	700kB～900kB
DIR-24-8	1/2	33MB
DIR-21-3-8	1/3	9MB
RAM 算法	1/2	7MB～12MB

RAM 算法是一种二级查表方法,其所需存储器大小可以用 $M_{RAM}=2^{16}*4+\left(\sum_{i=1}^{16}x[i]*2^i*4\right)$ (Bytes) 计算,而式(4)的值很小,可以忽略不计,因此本算法和 RAM 算法相比,所需存储器大小有大的改善,但增加一次访存.

本机制最大的优点是所需存储空间少,使得用 SRAM 实现容易,SRAM 与 DRAM 相比具有速度快、控制简单、不需动态刷新等特点,若 SRAM 的数量少,在价格上可与 DRAM 媲美. 本机制段表、偏移量表、压缩 NH 表均可用 64k*36 的 10ns 的 SRAM 实现,偏移量表和压缩 NH 表也可以在 SRAM 中顺序存放;其访存次数最小为 1,最大为 3,可以与 DIR-21-3-8 相媲美. 使用 FPGA 进行设计,其逻辑可以在 3 拍完成,主频为 40MHz. 若使用流水线设计,可以保证每拍出一个结果,查找速率可达 40Mpps,若 IP 分组为 40B 长,则可以满足 12.8Gbps 的链路速率.

4 结语

与文献[13,14]提出的位图(Bit Map)压缩技术相比,图论压缩方法是一种新方法;本文还针对不同的 m 取值建立两种压缩算法,尤其是压缩算法 1,其算法过程只需用结构与数组来实现,实现过程非常直观、简洁,总的时间复杂性为 $O(m^2)$,当 m 较小时,实现复杂性也不高,是一种比较实用算法,由于分段之后 m 的值一般较小,因此压缩算法 1 实用价值好;本章还给出了两种算法的正确性证明、复杂性和最优性讨论.

基于压缩 NH 表进行 IP 路由查找的方法属于硬件查找机制,具有查找速率快、所需存储空间少、硬件实现简单等特点,是一种理想的适合于 10Gbps 端口核心路由器的查找机制. 该方法已在实验室得到验证,其简化的硬件设计已用于核心路由器的设计,效果良好. 下一步的目标是研究将该算法应用于高速多维 IP 分组分类问题的方法.

参考文献:

[1] Y Rekhter, T Li. An Architecture for IP Address Allocation with CIDR

[S]. RFC 1518, 1993, http://www.ietf.org/rfc/rfc1518.txt.

[2] P Gupta, S Lin, N McKeown. Routing lookups in hardware at memory access speeds [A]. In Proceedings of INFOCOM [C], 1998: 1240 - 1247.

[3] A Brodnik, S Carlsson, M Degermark, S Pink. Small forwarding tables for fast routing lookups [A]. in Proceedings of ACM SIGCOMM [C]. 1997: 3 - 13.

[4] M Waldvogel, G Varghese, J Turner, B Plattner. Scalable high speed IP routing lookups [A]. in Proceedings of ACM SIGCOMM [C], 1997: 25 - 36.

[5] B Lampson, V Srinivasan, G Varghese. IP lookups using Multiway and Multicolumn search [A]. in Proceedings of INFORCOM [C], 1998: 1248 - 1256.

[6] V Srinivasan, G Varghese. Faster IP lookups using controlled prefix expansion [A]. in ACM Sigmetrics [C], 1998: 1 - 10.

[7] Michigan University, Merit Network. Internet performance measurement and analysis (IPMA) Project [Online] [DB/OL]. Available WWW: http://www.merit.edu/~ipma.

[8] Scott Bradner. Next generation routers overview [A]. Proceedings of Networld Interop 97 [C].

[9] C Labovitz, G Malan, F Jahanian. Internet routing instability [A]. in Proceedings of SIGCOMM 97 [C], 1997, 27(4): 115 - 126.

[10] T V Lakshman, D Stiliadis. High-speed policy-based packet forwarding using efficient multi-dimensional range matching [J]. ACM Computer Communication Review, 1998, 28(4): 203 - 214.

[11] Keith Sklower. A tree-based routing table for berkeley unix [A]. Proceedings of the Winter 1991 USENIX Conference [C], 1991: 93 - 99.

[12] A J McAuley, P Francis. Fast routing table lookup using CAMs [A]. IEEE INFOCOM '93 [C], 1993: 1382 - 91.

[13] 吴剑,陈修环,徐明伟,徐恪. 高性能安全路由器中快速路由查找算法的研究与实现 [J]. 电子学报, 2001, 11 月增刊: 123 - 125.

[14] N F Huang, S M Zhao, J Y Pan, C A Su. A fast IP routing lookup scheme for gigabit switching routers [A]. IEEE INFOCOM [C], 1999: 1429 - 1436.

[15] N F Huang, S M Zhao. A novel IP routing lookup scheme and hardware architecture for multigigabit switching routers [J]. IEEE JSAC, 1999, 17(6): 1093 - 1104.

作者简介:

彭元喜 男. 1966 年 4 月生于湖南省澧县. 博士研究生, 1988 年毕业于四川大学计算机科学系, 获学士学位, 1998 年毕业于国防科技大学计算机科学与技术系, 获硕士学位, 1998 年进入国防科技大学计算机学院读博士学位, 主要研究方向为快速路由查找、分组分类、交换、调度.

唐玉华 女. 1962 年 5 月生于江苏省南京市. 教授, 分别于 1983 年和 1986 年毕业于国防科技大学计算机科学与技术系, 获学士学位和硕士学位, 主要从事计算机网络和计算机体系结构方面的研究工作.

龚正虎 男. 1945 年 8 月生于湖南省长沙市. 教授, 博士生导师, 1970 年毕业于清华大学无线电电子学系, 曾为英国伯明翰大学和美国普渡大学访问学者. 目前主要从事计算机网络与通信, 分布计算, 网络协议工程等方面的研究与教学工作.