



(12) 发明专利申请

(10) 申请公布号 CN 102571599 A

(43) 申请公布日 2012. 07. 11

(21) 申请号 201210028320. 9

(22) 申请日 2012. 02. 08

(71) 申请人 福建三元达通讯股份有限公司

地址 350000 福建省福州市鼓楼区五凤街道  
软件大道 89 号福州软件园产业基地二  
期 7# 楼

(72) 发明人 林川 黄海峰 张媛

(74) 专利代理机构 福州市鼓楼区京华专利事务  
所(普通合伙) 35212

代理人 宋连梅

(51) Int. Cl.

H04L 12/56(2006. 01)

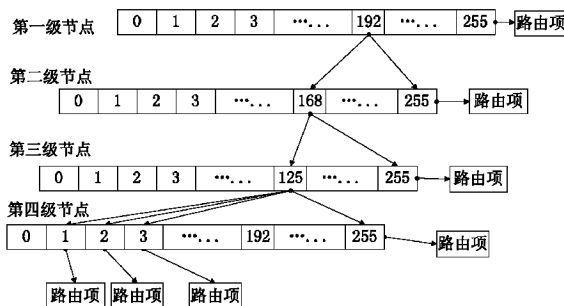
权利要求书 2 页 说明书 9 页 附图 4 页

(54) 发明名称

一种路由表项的快速存储方法

(57) 摘要

本发明提供一种路由表项的快速存储方法, 采用 256tire 树结构来配置路由表项, 该 256tire 树的树高为四, 四个节点均包括索引值为 0~255 的 256 个索引表项, 根节点的索引值为 255 的索引表项直接指向的路由表项表示默认路由; 而第二级节点的索引值为 255 的索引表项指向的是匹配中根节点后, 第二级节点都没有匹配的情况下, 应该返回的路由表项; 第三级节点的索引值为 255 的索引表项指向的是匹配中前两级节点后, 第三级节点都没有匹配中的情况下, 应该返回的路由表项; 第四级节点的索引值为 255 的索引表项指向的是匹配中前三级节点后, 第四级节点都没有匹配中的情况下, 应该返回的路由表项。本发明不仅路由查找都会很快速, 提高了路由查找的效率, 从而提高路由器的转发性能, 还能节省内存空间。



1. 一种路由表项的快速存储方法,其特征在于:采用 256tire 树结构来配置路由表项,该 256tire 树的树高为四,即包括根节点、第二级节点,第三级节点,第四级节点,所述根节点、第二级节点,第三级节点,第四级节点均包括索引值为 0~255 的 256 个索引表项,所述根节点的索引值为 255 的索引表项直接指向的路由表项表示默认路由;而第二级节点的索引值为 255 的索引表项指向的是匹配中根节点后,第二级节点都没有匹配的情况下,应该返回的路由表项;第三级节点的索引值为 255 的索引表项指向的是匹配中前两级节点后,第三级节点都没有匹配中的情况下,应该返回的路由表项;第四级节点的索引值为 255 的索引表项指向的是匹配中前三级节点后,第四级节点都没有匹配中的情况下,应该返回的路由表项。

2. 如权利要求 1 所述的一种路由表项的快速存储方法,其特征在于:将任意一个目的 ip 地址,都用 4 个一字节的数值表示;该 4 个一字节的数值分别称为 ip 地址的第一字节,第二字节,第三字节和第四字节;所述路由表项的快速存储方法包括 256trie 树生成的过程,该 256trie 树生成的过程包括下述步骤:

步骤 (1)、判断配置的路由表项的类型:

a、如果配置的是默认路由,则直接把根节点的索引值为 255 的索引表项指向该默认路由表项,无需进入步骤 (2);

b、如果是掩码长度小于等于 8 的路由表项,需要设置到根节点,所有索引值和目的 ip 地址前缀匹配的,且前缀匹配长度为掩码长度的索引表项都要设置路由表项;设置路由表项的过程按步骤 (2) 进行;

c、如果是掩码长度大于 8 小于等于 16 的路由表项,先取目的 ip 的第一字节,算出根节点的索引值,然后取出目的 ip 地址的第二字节,找到和第二字节前缀匹配,并且匹配长度是掩码减 8 的所有第二级节点的索引表项,这些第二级节点的索引表项和刚才算出的索引值对应的索引表项建立关联,同时按步骤 (2) 设置这个路由表项;

d、如果是掩码长度大于 16 小于等于 24 的路由表项,先取目的 ip 地址的前两字节,算出根节点的索引值和第二级的索引值,该根节点的索引值和第二级的索引值分别等于目的 ip 地址的第一字节,第二字节的值,把根节点中与根节点的索引值对应的索引表项关联到第二级节点,把第二级节点中与第二级的索引值对应的索引表项关联到第三级节点,然后,取出目的 ip 地址的第三字节,找到和第三字节前缀匹配,并且长度是掩码长度减 16 的所有第三级节点的索引表项,这些第三级节点的索引表项,和所述与第二级的索引值对应的索引表项建立关联,同时按步骤 (2) 设置路由表项;

e、如果掩码长度大于 24 小于等于 32 的路由表项,先取目的 ip 地址的前三字节,算出根节点的索引值、第二级的索引值和第三级的索引值,该根节点的索引值、第二级的索引值和第三级的索引值分别等于目的 ip 地址的第一字节,第二字节的值、第三字节的值,把根节点中与根节点的索引值对应的索引表项关联到第二级节点,把第二级节点中与第二级的索引值对应的索引表项关联到第三级节点,把第三级节点中与第三级的索引值对应的索引表项关联到第四级节点,然后,取出目的 ip 地址的第四字节,找到和第四字节前缀匹配,并且长度是掩码长度减 24 的所有第四级索引表项,这些第四级索引表项和所述的第三级的索引值对应的索引表项建立关联,同时按步骤 (2) 设置路由表项;

步骤 (2)、设置路由表项:对于前三级节点,除了默认路由以外,要关联路由表项,上一

级节点都是和下一级节点的索引值为 255 的索引表项建立关联,然后将该索引的索引值为 255 的索引表项指向这个路由表项;第四级节点表项则是直接指向步骤(1)对应的所述路由表项;在设置路由表项时要先看是否已经关联了路由表项了,如果该索引表项已经指向了一个路由表项,则比较这两个路由表项中掩码长度,该索引表项与掩码长度长的那个路由表项关联;

如果上一级节点要关联的下一级节点没有创建,则需要先创建下一级节点。

3. 如权利要求 2 所述的一种路由表项的快速存储方法,其特征在于:该路由表项的快速存储方法还包括 256tire 树查找路由的过程,所述 256tire 树查找路由的过程为:

步骤 1、首先取 ip 地址的第一字节,在根节点中匹配索引值等于该第一字节的索引表项,即看根节点中索引值等于该第一字节的索引表项是否关联到第二级节点,如果否,说明未匹配中,则返回根节点中的索引值为 255 的索引表项指向的路由表项,如果根节点中的索引值为 255 的索引表项没有指向任何路由表项,则返回路由查找失败;如果是,说明匹配中,则转入步骤 2;

步骤 2、取 ip 地址的第二字节,看在步骤 1 根节点的索引值等于该第一字节的索引表项所关联的第二级节点中,索引值和 ip 地址的第二字节对应的第二级索引表项是否和第三级节点建立了关联;如果建立了关联则转步骤 3,如果没有建立关联,则返回第二级节点的索引值为 255 的索引表项指向的路由表项,如果第二级节点的索引值为 255 的索引表项没有指向路由表项,则返回根节点的索引值为 255 的索引表项指向的路由表项,如果根节点的索引值为 255 的索引表项没有指向路由表项,则返回查找失败;

步骤 3、取 ip 地址的第三字节,看步骤 2 中第二级索引表项所关联的第三级节点中,索引值和该 ip 地址第三字节相同的第三级索引表项是否关联到第四级节点;如果建立了关联则转步骤 4,如果没有建立关联,则返回第三级节点的索引值为 255 的索引表项指向的路由表项,如果第三级节点的索引值为 255 的索引表项没有指向路由表项,则返回第二级节点的索引值为 255 的索引表项指向的路由表项,如果第二级节点的索引值为 255 的索引表项没有指向路由表项,则返回根节点索引值为 255 的索引表项指向的路由表项,如果根节点的索引值为 255 的索引表项没有指向路由表项,返回查找失败;

步骤 4、取 ip 地址的第四字节,看步骤 3 中第三级索引表项所关联的第四级节点中,和 ip 地址第四字节相同的索引值所对应的第四级索引表项是否关联到第四级节点;如果建立了关联,返回第四级节点对应的索引表项指向的路由表项;如果没有建立关联,则返回第四级节点的索引值为 255 的索引表项指向的路由表项,如果四级索引的索引值为 255 的索引表项没有指向路由表项,则返回第三级节点的索引值为 255 的索引表项指向的路由表项,如果第三级节点的索引值为 255 的索引表项没有指向路由表项,则返回第二级节点的索引值为 255 的索引表项指向的路由表项,如果第二级节点的索引值为 255 的索引表项没有指向路由表项,则返回根节点的索引值为 255 的索引表项所指向的路由表项,如果根节点没有指向路由表项,则返回查找失败;

其中,所述的关联是指上一级节点中,与某个索引值对应的索引表项指向下一级节点,或者指向路由表项。

## 一种路由表项的快速存储方法

### 【技术领域】

[0001] 本发明涉及一种通信技术,特别涉及一种路由表项的存储方法,不仅路由查找都会很快,提高了路由查找的效率,从而提高路由器的转发性能,还能节省内存空间。

### 【背景技术】

[0002] 由于 Internet 的飞速发展,网络用户数目的增长,多媒体网络应用日益广泛,网络流量呈爆炸式的增长趋势。Internet 若要想继续提供较好的服务,要求核心路由器每秒能转发几百万个以上的分组,快速路由查找技术成为路由器报文转发的瓶颈。因此如何实现高速路由表的查找和更新是研究的难点。

[0003] 1、先了解几个概念:

[0004] 路由:是指路由器从一个接口上收到数据包,根据数据包的目的地址进行定向并转发到另一个接口的过程。

[0005] 路由表:在计算机网络中,路由表或称路由择域信息库(RIB)是一个存储在路由器或者联网计算机中的电子表格(文件)或类数据库。路由表存储着指向特定网络地址的路径(在有些情况下,十进制的数据也附属在路径上)。路由表中含有网络周边的拓扑信息。路由表建立的主要目标是为了实现路由协议和静态路由选择。

[0006] 2、路由表的主要工作及表项类型:

[0007] 路由器的主要工作就是为经过路由器的每个数据报寻找一条最佳传输路径,并将该数据有效地传送到目的站点。由此可见,选择最佳路径的策略即路由算法是路由器的关键所在。为了完成这项工作,在路由器中保存着各种传输路径的相关数据——路由表(Routing Table),供路由选择时使用,表中包含的信息决定了数据转发的策略。打个比方,路由表就像我们平时使用的地图一样,标识着各种路线,路由表中保存着子网的标志信息、网上路由器的个数和下一个路由器的名字等内容。路由表可以由系统管理员固定设置好的,也可以由系统动态修改,可以由路由器自动调整,也可以由主机控制。路由表包括静态路由表和动态路由表:

[0008] (1) 静态路由表:由系统管理员事先设置好固定的路由表称之为静态(static)路由表,一般是在系统安装时就根据网络的配置情况预先设定的,它不会随未来网络结构的改变而改变。

[0009] (2) 动态路由表:动态(Dynamic)路由表是路由器根据网络系统的运行情况而自动调整的路由表。路由器根据路由选择协议(Routing Protocol)提供的功能,自动学习和记忆网络运行情况,在需要时自动计算数据传输的最佳路径。

[0010] 路由器通常依靠所建立及维护的路由表来决定如何转发。路由表能力是指路由表内所容纳路由表项数量的极限。由于 Internet 上执行 BGP 协议的路由器通常拥有数十万条路由表项,所以该项目也是路由器能力的重要体现。路由表项如下:

[0011] 首先,路由表的每个项的目的字段含有目的网络前缀。其次,每个项还有一个附加字段,还有用于指定网络前缀位数的子网掩码(address mask)。第三,当下一跳字段代表路

由器时,下一跳字段的值使用路由的 IP 地址。

[0012] 理解网际网络中可用的网络地址(或网络 ID)有助于路由决定。这些知识是从称为路由表的数据库中获得的。路由表是一系列称为路由的项,其中包含有关网际网络的网络 ID 位置信息。路由表不是对路由器专用的。主机(非路由器)也可能有用来决定优化路由的路由表。路由表中的每一项都被看作是一个路由,并且属于下列任意类型:

[0013] (1) 网络路由:网络路由提供到网际网络中特定网络 ID 的路由。

[0014] (2) 主路由:主路由提供到网际网络地址(网络 ID 和节点 ID)的路由。主路由通常用于将自定义路由创建到特定主机以控制或优化网络通信。

[0015] (3) 默认路由:如果在路由表中没有找到其他路由,则使用默认路由。例如,如果路由器或主机不能找到目标的网络路由或主路由,则使用默认路由。默认路由简化了主机的配置。使用单个默认的路由来转发带有在路由表中未找到的目标网络或网际网络地址的所有数据包,而不是为网际网络中所有的网络 ID 配置带有路由的主机。

[0016] (4) 无类路由:即可以在每个网络段中使用不相同的子网掩码,RIPv2、EIGRP、OSPF 和 BGP 等是一些比较新的路由选择协议,它们在路由更新过程中,将网络掩码与路径一起广播出去,这时网络掩码也称为前缀屏蔽或前缀。例如,如果 C 类 IP 地址 192.168.1.0 的网络掩码为 255.255.255.0,可标识为 192.168.1.0/24。由于在路由器之间传送掩码(前缀),因而没有必要判断地址类型和缺省掩码,这就是无类地址及无类路由选择,也是目前 Internet 上所基于的路由选择协议。

[0017] 对于有类路由协议:RIP、IGRP 等。路由控制包里面只有 IP 地址,没有掩码。路由协议根据 IP 地址的具体值,按规定划分网络 ID 和主机 ID。这就是有类路由(Classful Routing)。Classful Routing 规定,IP 地址有五种分类:A,B,C,D,E。

[0018] A 类地址:前 8 位是 0~127 的 IP 地址,它们的网络地址是前 8 位,主机地址是后 24 位。

[0019] B 类地址:前 8 位是 128~191 的 IP 地址,它们的网络地址是前 16 位,主机地址是后 16 位。

[0020] C 类地址:前 8 位是 192~223 的 IP 地址,他们的网络地址是前 24 位,主机地址是后 8 位。

[0021] D 类和 E 类地址:前 8 位是 224~255 之间的地址,D 类用于组播,E 类用于科学实验。

[0022] 而对于无类的路由协议:OSPF、IS-IS、RIPv2 等,路由控制包里面除了 IP 地址,还必须包含一个掩码与其匹配。掩码的作用是划分出网络 ID 和主机 ID,这样的划分更灵活多变。

[0023] 3、现有技术及解决方案

[0024] 目前,交换机,路由器实现路由的方法多种多样,有些通过硬件的三层转发表项实现路由更新,有些通过软件实现,软件实现路由的方法也很多,采用 Trie 树的方法,采用 HASH 的方法,不过,采用硬件 3 层转发的成本较高,采用软件路由的,算法都比较复杂,实现起来较为困难。

[0025] Trie 树,又称单词查找树或键树,是一种树形结构,是一种哈希树的变种。典型应用是用于统计和排序大量的字符串(但不仅限于字符串),所以经常被搜索引擎系统用于

文本词频统计。它的优点是：最大限度地减少无谓的字符串比较，查询效率比哈希表高。它有 3 个基本特性：

【0026】 1) 根节点不包含字符，除根节点外每一个节点都只包含一个字符；

【0027】 2) 从根节点到某一节点，路径上经过的字符连接起来，为该节点对应的字符串；

【0028】 3) 每个节点的所有子节点包含的字符都不相同。

【0029】 如图 1 所示，为一种 Trie 树结构的示意图，如图所示，该 trie 树存有 abc、d、da、dda 四个字符串，如果是字符串会在节点的尾部进行标记。没有后续字符的 branch 分支指向 NULL（即无效数据）。

【0030】 如图 2 所示，Trie 树又包括 256trie 树，图 2 为 256trie 树结构的示意图，256trie 树是字符集为 {0, 1, 2... 255} 的 Trie 树，可以用来做路由表查找；对于 ipv4 路由，目的 ip 可以分为 4 个字节的字符，其中每个字节的值，都在 0-255 的范围内；

【0031】 目前的路由查找算法，有的需要硬件配合，有的算法十分复杂，软件实现比较困难，经检索，利用 Trie 树来实现路由的方法有：

【0032】 公开日为 2009-11-11，公开号为 101577662 的中国发明专利申请 [200810096906. 2] 公开了最长前缀匹配方法和装置，该方法包括：A. 读取一个搜索节点；B. 确定读出的搜索节点的偏移量域是否指示上一级别的节点内存在匹配的前缀，如果存在，将上一级别的节点内指向叶子节点数组的指针加上该搜索节点的偏移量域，更新当前最佳匹配指针，并执行步骤 C；如果不存在，执行步骤 C；C 确定该搜索节点的分支指示域和搜索关键字的对应比特匹配时，确定该搜索节点是否存在子节点；D. 确定该搜索节点不存在子节点时，读取该搜索节点的内部位图，根据内部位图和搜索节点内指向叶子节点数组的指针，计算该搜索节点内存在的最长匹配前缀，更新当前最佳匹配指针，计算当前最佳匹配指针对应的叶子节点的地址。该方法可以提高查询速度。该算法是对 Trie 算法的一种改进，起到了提高路由查找效率的作用，但是该算法比较复杂。

【0033】 公开日为 2006-06-14，公开号为 1787477 的中国发明专利申请 [200410009995. 4] 公开了一种 IPv6 路由表查找方法，包括以下步骤：a) 将整个 IPv6 路由表分割成至少两个前缀子集；b) 对前缀数目较多或前缀长度较长的前缀子集采用索引表进行检索，对前缀数目较少且前缀长度较短的前缀子集，利用多分支 Trie 树进行查找；以及 c) 在全局范围内利用至少两个微处理器模块来对每一部分前缀子集进行并行查找，在局部范围内利用路由预计算进行查找。该方法能十分有效地支持 IPv6 路由查找，不依赖于 TCAM，可以在通用硬件平台上实现。

【0034】 但申请人没有在公开的文献中检索到利用 256Trie 树算法的路由方法，据了解，有些企业目前的路由算法也有采用 256 树的办法，只是需要创建很多节点，会过度的使用内存空间，而且软件算法实现复杂，不支持无类路由，因此使用不是很广泛。

## 【发明内容】

【0035】 本发明要解决的技术问题，在于提供一种路由表项存储方法，对现有的 256Trie 树的方法进行改进，能够仅依赖于软件实现路由快速存储和快速查找，达到提高转发性能的目的，同时节约内存，而且软件算法实现简单，能符合一个字节 8 比特的特性，方便程序员编程。

[0036] 本发明是这样实现的：一种路由表项的快速存储方法，其是采用 256tire 树结构来配置路由表项，该 256tire 树的树高为四，即包括根节点、第二级节点，第三级节点，第四级节点，所述根节点、第二级节点，第三级节点，第四级节点均包括索引值为 0 ~ 255 的 256 个索引表项，所述根节点的索引值为 255 的索引表项直接指向的路由表项表示默认路由；而第二级节点的索引值为 255 的索引表项指向的是匹配中根节点后，第二级节点都没有匹配的情况下，应该返回的路由表项；第三级节点的索引值为 255 的索引表项指向的是匹配中前两级节点后，第三级节点都没有匹配中的情况下，应该返回的路由表项；第四级节点的索引值为 255 的索引表项指向的是匹配中前三级节点后，第四级节点都没有匹配中的情况下，应该返回的路由表项。

[0037] 上述路由表项的快速存储方法进一步是将任意一个目的 ip 地址，都用 4 个一字节的数值表示；该 4 个一字节的数值分别称为 ip 地址的第一字节，第二字节，第三字节和第四字节；所述路由表项的快速存储方法包括 256trie 树生成的过程，该 256trie 树生成的过程包括下述步骤：

[0038] 步骤 (1)、判断配置的路由表项的类型：

[0039] a、如果配置的是默认路由，则直接把根节点的索引值为 255 的索引表项指向该默认路由表项，无需进入步骤 (2)；

[0040] b、如果是掩码长度小于等于 8 的路由表项，需要设置到根节点，所有索引值和目的 ip 地址前缀匹配的，且前缀匹配长度为掩码长度的索引表项都要设置路由表项；设置路由表项的过程按步骤 (2) 进行；

[0041] c、如果是掩码长度大于 8 小于等于 16 的路由表项，先取目的 ip 的第一字节，算出根节点的索引值，然后取出目的 ip 地址的第二字节，找到和第二字节前缀匹配，并且匹配长度是掩码减 8 的所有第二级节点的索引表项，这些第二级节点的索引表项和刚才算出的索引值对应的索引表项建立关联，同时按步骤 (2) 设置这个路由表项；

[0042] d、如果是掩码长度大于 16 小于等于 24 的路由表项，先取目的 ip 地址的前两字节，算出根节点的索引值和第二级的索引值，该根节点的索引值和第二级的索引值分别等于目的 ip 地址的第一字节，第二字节的值，把根节点中与根节点的索引值对应的索引表项关联到第二级节点，把第二级节点中与第二级的索引值对应的索引表项关联到第三级节点，然后，取出目的 ip 地址的第三字节，找到和第三字节前缀匹配，并且长度是掩码长度减 16 的所有第三级节点的索引表项，这些第三级节点的索引表项，和所述与第二级的索引值对应的索引表项建立关联，同时按步骤 (2) 设置路由表项；

[0043] e、如果掩码长度大于 24 小于等于 32 的路由表项，先取目的 ip 地址的前三字节，算出根节点的索引值、第二级的索引值和第三级的索引值，该根节点的索引值、第二级的索引值和第三级的索引值分别等于目的 ip 地址的第一字节，第二字节的值、第三字节的值，把根节点中与根节点的索引值对应的索引表项关联到第二级节点，把第二级节点中与第二级的索引值对应的索引表项关联到第三级节点，把第三级节点中与第三级的索引值对应的索引表项关联到第四级节点，然后，取出目的 ip 地址的第四字节，找到和第四字节前缀匹配，并且长度是掩码长度减 24 的所有第四级索引表项，这些第四级索引表项和所述的第三级的索引值对应的索引表项建立关联，同时按步骤 (2) 设置路由表项；

[0044] 步骤 (2)、设置路由表项：对于前三级节点，除了默认路由以外，要关联路由表项，

上一级节点都是和下一级节点的索引值为 255 的索引表项建立关联,然后将该索引的索引值为 255 的索引表项指向这个路由表项;第四级节点表项则是直接指向步骤(1)对应的所述路由表项;在设置路由表项时要先看是否已经关联了路由表项了,如果该索引表项已经指向了一个路由表项,则比较这两个路由表项中掩码长度,该索引表项与掩码长度长的那个路由表项关联;

[0045] 如果上一级节点要关联的下一级节点没有创建,则需要先创建下一级节点。

[0046] 进一步地,该路由表项的快速存储方法还包括 256tire 树查找路由的过程,所述 256tire 树查找路由的过程为:

[0047] 步骤 1、首先取 ip 地址的第一字节,在根节点中匹配索引值等于该第一字节的索引表项,即看根节点中索引值等于该第一字节的索引表项是否关联到第二级节点,如果否,说明未匹配中,则返回根节点中的索引值为 255 的索引表项指向的路由表项,如果根节点中的索引值为 255 的索引表项没有指向任何路由表项,则返回路由查找失败;如果是,说明匹配中,则转入步骤 2;

[0048] 步骤 2、取 ip 地址的第二字节,看在步骤 1 根节点的索引值等于该第一字节的索引表项所关联的第二级节点中,索引值和 ip 地址的第二字节对应的第二级索引表项是否和第三级节点建立了关联;如果建立了关联则转步骤 3,如果没有建立关联,则返回第二级节点的索引值为 255 的索引表项指向的路由表项,如果第二级节点的索引值为 255 的索引表项没有指向路由表项,则返回根节点的索引值为 255 的索引表项指向的路由表项,如果根节点的索引值为 255 的索引表项没有指向路由表项,则返回查找失败;

[0049] 步骤 3、取 ip 地址的第三字节,看步骤 2 中第二级索引表项所关联的第三级节点中,索引值和该 ip 地址第三字节相同的第三级索引表项是否关联到第四级节点;如果建立了关联则转步骤 4,如果没有建立关联,则返回第三级节点的索引值为 255 的索引表项指向的路由表项,如果第三级节点的索引值为 255 的索引表项没有指向路由表项,则返回第二级节点的索引值为 255 的索引表项指向的路由表项,如果第二级节点的索引值为 255 的索引表项没有指向路由表项,则返回根节点索引值为 255 的索引表项指向的路由表项,如果根节点的索引值为 255 的索引表项没有指向路由表项,返回查找失败;

[0050] 步骤 4、取 ip 地址的第四字节,看步骤 3 中第三级索引表项所关联的第四级节点中,和 ip 地址第四字节相同的索引值所对应的第四级索引表项是否关联到第四级节点;如果建立了关联,返回第四级节点对应的索引表项指向的路由表项;如果没有建立关联,则返回第四级节点的索引值为 255 的索引表项指向的路由表项,如果四级索引的索引值为 255 的索引表项没有指向路由表项,则返回第三级节点的索引值为 255 的索引表项指向的路由表项,如果第三级节点的索引值为 255 的索引表项没有指向路由表项,则返回第二级节点的索引值为 255 的索引表项指向的路由表项,如果第二级节点的索引值为 255 的索引表项没有指向路由表项,则返回根节点的索引值为 255 的索引表项指向的路由表项,如果根节点没有指向路由表项,则返回查找失败;

[0051] 其中,所述的关联是指上一级节点中,与某个索引值对应的索引表项指向下一级节点,或者指向路由表项。

[0052] 本发明具有如下优点:改进的 256trie 是根据静态路由配置以及动态路由生成的路由表项生成的,主要是以目的网络地址为索引,最后都指向一个路由表项;每个路由表项



都包含目的 ip、网络掩码长度、下一条的 ip 或者出口。该方法能符合一个字节 8 比特的特性,方便软件编程;实质上是通过改进 Mtrie 算法,在 Mtrie 树的每级都用 255 索引指向这级索引值都匹配不中的情况下的路由表项,这样不用每次都匹配到第四级节点,通过该办法,能大量节省创建节点的数量,从而达到节约内存的效果;采用了每次路由表项前都比较下之前表项和现在表项的掩码长度,保证每个节点指向的路由表项都是掩码最长的,实现节点的最长前缀匹配,很好的支持了无类路由。

### 【附图说明】

[0053] 下面参照附图结合实施例对本发明作进一步的说明。

[0054] 图 1 为现有技术中一种 Trie 树结构的示意图。

[0055] 图 2 为现有技术中一种 256Trie 树结构的示意图。

[0056] 图 3 为本发明一种 256Trie 树结构的示意图。

[0057] 图 4 至图 6 为本发明一实施例中 256Trie 树结构的路由表项的创建过程示意图。

[0058] 图 7 为本发明的查找路由的过程的一实施例的 256tire 树结构示意图。

[0059] 图 8 为本发明的查找路由的过程的另一实施例的 256tire 树结构示意图。

[0060] 图 9 为本发明的查找路由的过程的再一实施例的 256tire 树结构示意图。

### 【具体实施方式】

[0061] 本发明要解决的技术问题,在于提供一种路由表项存储方法,对现有的 256Trie 树的方法进行改进,能够仅依赖于软件实现路由快速存储和快速查找,达到提高转发性能的目的,同时节约内存,而且软件算法实现简单,能符合一个字节 8 比特的特性,方便程序员编程。

[0062] 如图 3 所示,本发明方法是采用 256tire 树结构来配置路由表项,该 256tire 树的树高为四,即包括根节点、第二级节点,第三级节点,第四级节点,所述根节点、第二级节点,第三级节点,第四级节点均包括索引值为 0 ~ 255 的 256 个索引表项,所述根节点的索引值为 255 的索引表项直接指向的路由表项表示默认路由;而第二级节点的索引值为 255 的索引表项指向的是匹配中根节点后,第二级节点都没有匹配的情况下,应该返回的路由表项;第三级节点的索引值为 255 的索引表项指向的是匹配中前两级节点后,第三级节点都没有匹配中的情况下,应该返回的路由表项;第四级节点的索引值为 255 的索引表项指向的是匹配中前三级节点后,第四级节点都没有匹配中的情况下,应该返回的路由表项。

[0063] 上述路由表项的快速存储方法进一步是将任意一个目的 ip 地址,都用 4 个一字节的数值表示;该 4 个一字节的数值分别称为 ip 地址的第一字节,第二字节,第三字节和第四字节;所述路由表项的快速存储方法包括 256trie 树生成的过程,该 256trie 树生成的过程包括下述步骤:

[0064] 步骤 (1)、判断配置的路由表项的类型:

[0065] a、如果配置的是默认路由,则直接把根节点的索引值为 255 的索引表项指向该默认路由表项,无需进入步骤 (2);

[0066] b、如果是掩码长度小于等于 8 的路由表项,需要设置到根节点,所有索引值和目的 ip 地址前缀匹配的,且前缀匹配长度为掩码长度的索引表项都要设置路由表项;设置路

由表项的过程按步骤 (2) 进行；

[0067] c、如果是掩码长度大于 8 小于等于 16 的路由表项，先取目的 ip 的第一字节，算出根节点的索引值，然后取出目的 ip 地址的第二字节，找到和第二字节前缀匹配，并且匹配长度是掩码减 8 的所有第二级节点的索引表项，这些第二级节点的索引表项和刚才算出的索引值对应的索引表项建立关联，同时按步骤 (2) 设置这个路由表项；

[0068] d、如果是掩码长度大于 16 小于等于 24 的路由表项，先取目的 ip 地址的前两字节，算出根节点的索引值和第二级的索引值，该根节点的索引值和第二级的索引值分别等于目的 ip 地址的第一字节，第二字节的值，把根节点中与根节点的索引值对应的索引表项关联到第二级节点，把第二级节点中与第二级的索引值对应的索引表项关联到第三级节点，然后，取出目的 ip 地址的第三字节，找到和第三字节前缀匹配，并且长度是掩码长度减 16 的所有第三级节点的索引表项，这些第三级节点的索引表项，和所述与第二级的索引值对应的索引表项建立关联，同时按步骤 (2) 设置路由表项；

[0069] e、如果掩码长度大于 24 小于等于 32 的路由表项，先取目的 ip 地址的前三字节，算出根节点的索引值、第二级的索引值和第三级的索引值，该根节点的索引值、第二级的索引值和第三级的索引值分别等于目的 ip 地址的第一字节，第二字节的值、第三字节的值，把根节点中与根节点的索引值对应的索引表项关联到第二级节点，把第二级节点中与第二级的索引值对应的索引表项关联到第三级节点，把第三级节点中与第三级的索引值对应的索引表项关联到第四级节点，然后，取出目的 ip 地址的第四字节，找到和第四字节前缀匹配，并且长度是掩码长度减 24 的所有第四级索引表项，这些第四级索引表项和所述的第三级的索引值对应的索引表项建立关联，同时按步骤 (2) 设置路由表项；

[0070] 步骤 (2)、设置路由表项：对于前三级节点，除了默认路由以外，要关联路由表项，上一级节点都是和下一级节点的索引值为 255 的索引表项建立关联，然后将该索引的索引值为 255 的索引表项指向这个路由表项；第四级节点表项则是直接指向步骤 (1) 对应的所述路由表项；在设置路由表项时要先看是否已经关联了路由表项了，如果该索引表项已经指向了一个路由表项，则比较这两个路由表项中掩码长度，该索引表项与掩码长度长的那个路由表项关联；

[0071] 如果上一级节点要关联的下一级节点没有创建，则需要先创建下一级节点。

[0072] 现分情况举例说明该 256trie 树生成的过程：

[0073] 实施例一

[0074] 请举例说明上述 a 种情况，

[0075] 例如用户配置了默认路由：则根节点索引值为 255 的索引表项直接指向了这个默认路由项。

[0076] 实施例二

[0077] 请举例说明上述 b 种情况

[0078] 如果配置了目的地址 / 掩码为：10. 0. 0. 0/8 的路由表项，则 256trie 树中根节点的索引值为 10 的索引表项指向第二级节点，而第二级节点中索引值为 255 的索引表项指向该路由项。

[0079] 实施例三

[0080] 如图 4 所示，一开始路由表只有根节点，且根节点不关联任何路由表项；这时，用

户配置了一条路由目的 ip 地址和掩码为 :192. 168. 12. 23/15, 则先匹配中根节点索引值为 192 的索引表项, 然后看 ip 地址的第二字节为 168, 掩码长度 15, 也就是找到第二级节点索引值为 168、前缀匹配长度为 7 的两个索引表项, 也就是索引值为 168 和 169 的两个索引表项, 然后建立关联, 并且同时, 第二级节点也要和路由项建立关联, 按照上述步骤 (1) 中的 d 种情况介绍的方法和通过第三级节点的 255 项关联路由表项 ; 该路由表项设置后 256Trie 树变为如图 5 所示的状态。如图 6 所示, 如果此时用户再配置了 192. 168. 12. 23/9 这个路由表项, 则第二级节点索引值为 128-254 的索引表项都要被设置成该路由表项 ( 图中实线箭头部分 ), 除了索引值为 168、169 两个索引表项 ( 图中虚线箭头部分 ), 因为它们所指向的下一级节点的索引值为 255 的索引表项所指向的路由表项的掩码长度大于 9, 所以索引值为 168、169 两个索引表项不会被改变。

#### [0081] 实施例四

[0082] 用户配置了一条网络地址和掩码为 :192. 168. 125. 0/24 的路由表项, 这时如图 7 所示, 根节点索引值为 192 的索引表项, 指向第二级节点, 第二级节点的索引值为 168 的索引表项指向第三级节点, 第三级节点索引值为 125 的索引表项指向第四级节点, 第四级节点的索引值为 255 的索引表项指向该路由项。图中虚线箭头表示的应该是原先配置的路由项, 生成的 256trie 树后, 虚线箭头链接的索引表项关联不变。

[0083] 本发明方法还包括 256tire 树查找路由的过程, 所述 256tire 树查找路由的过程为 :

[0084] 步骤 1、首先取 ip 地址的第一字节, 在根节点中匹配索引值等于该第一字节的索引表项, 即看根节点中索引值等于该第一字节的索引表项是否关联到第二级节点, 如果否, 说明未匹配中, 则返回根节点中的索引值为 255 的索引表项指向的路由表项, 如果根节点中的索引值为 255 的索引表项没有指向任何路由表项, 则返回路由查找失败 ; 如果是, 说明匹配中, 则转入步骤 2 ;

[0085] 步骤 2、取 ip 地址的第二字节, 看在步骤 1 根节点的索引值等于该第一字节的索引表项所关联的第二级节点中, 索引值和 ip 地址的第二字节对应的第二级索引表项是否和第三级节点建立了关联 ; 如果建立了关联则转步骤 3, 如果没有建立关联, 则返回第二级节点的索引值为 255 的索引表项指向的路由表项, 如果第二级节点的索引值为 255 的索引表项没有指向路由表项, 则返回根节点的索引值为 255 的索引表项指向的路由表项, 如果根节点的索引值为 255 的索引表项没有指向路由表项, 则返回查找失败 ;

[0086] 步骤 3、取 ip 地址的第三字节, 看步骤 2 中第二级索引表项所关联的第三级节点中, 索引值和该 ip 地址第三字节相同的第三级索引表项是否关联到第四级节点 ; 如果建立了关联则转步骤 4, 如果没有建立关联, 则返回第三级节点的索引值为 255 的索引表项指向的路由表项, 如果第三级节点的索引值为 255 的索引表项没有指向路由表项, 则返回第二级节点的索引值为 255 的索引表项指向的路由表项, 如果第二级节点的索引值为 255 的索引表项没有指向路由表项, 则返回根节点索引值为 255 的索引表项指向的路由表项, 如果根节点的索引值为 255 的索引表项没有指向路由表项, 返回查找失败 ;

[0087] 步骤 4、取 ip 地址的第四字节, 看步骤 3 中第三级索引表项所关联的第四级节点中, 和 ip 地址第四字节相同的索引值所对应的第四级索引表项是否关联到第四级节点 ; 如果建立了关联, 返回第四级节点对应的索引表项指向的路由表项 ; 如果没有建立关联, 则

返回第四级节点的索引值为 255 的索引表项值指向的路由表项, 如果四级索引的索引值为 255 的索引表项没有指向路由表项, 则返回第三级节点的索引值为 255 的索引表项指向的路由表项, 如果第三级节点的索引值为 255 的索引表项没有指向路由表项, 则返回第二级节点的索引值为 255 的索引表项指向的路由表项, 如果第二级节点的索引值为 255 的索引表项没有指向路由表项, 则返回根节点的索引值为 255 的索引表项所指向的路由表项, 如果根节点没有指向路由表项, 则返回查找失败;

[0088] 其中, 所述的关联是指上一级节点中, 与某个索引值对应的索引表项指向下一级节点, 或者指向路由表项。

[0089] 为了更好地说明, 下面举几个例子来说明本发明 256trie 树查找路由的过程;

[0090] 实施例五

[0091] 以 ip 192. 168. 125. 2 为例, 如果 256Trie 树的结构是如图 3 所示的结构, 则索引过程如图 8 ; 经过对 ip 地址 192. 168. 125. 2 的四次索引就查到了该目的 ip 的路由项。图中虚线箭头表示的应该是原先配置的路由项, 生成的 256trie 树后, 虚线箭头链接的索引表项关联不变。(此实施例最好能按前述步骤 1-4 进行较详细地描述)

[0092] 实施例六

[0093] 如图 9 所示, 以 ip 地址 192. 168. 135. 2 为例, 前两级节点的匹配可参照实施五中前两级节点的匹配, 匹配第三级节点时, 第三级节点中索引值为 135 的索引项没有匹配中, 于是匹配第三级节点中索引值为 255 的索引项。图中虚线箭头表示的应该是原先配置的路由项, 生成的 256trie 树后, 虚线箭头链接的索引表项关联不变。

[0094] 实施例七

[0095] 再如图 9 所示, 以 ip 地址 192. 11. 11. 3 为例, 第一级节点匹配中索引值为 192 的索引项后, 索引值为 192 的索引项没有关联到第二级节点索引值为 11 的索引项, 所以只能匹配到第二级节点索引值为 255 的索引项, 需要注意的是, 如果第二级节点索引值为 255 的索引项没有指向路由项, 则应该匹配第一级节点匹配中索引值为 255 的索引项指向的路由项。

[0096] 本发明的 256trie 是根据静态路由配置以及动态路由生成的路由表项生成的, 主要是以目的网络地址为索引, 最后都指向一个路由表项; 每个路由表项都包含目的 ip、网络掩码长度、下一条的 ip 或者出口。该方法能符合一个字节 8 比特的特性, 方便软件编程; 实质上是通过 Mtrie 算法改进, 在 Mtrie 树的每级都用 255 索引指向这级索引值都匹配不中的情况下的路由表项, 这样不用每次都匹配到第四级节点, 通过该办法, 能大量节省创建节点的数量, 从而达到节约内存的效果; 采用了每次路由表项前都比较下之前表项和现在表项的掩码长度, 保证每个节点指向的路由表项都是掩码最长的, 实现节点的最长前缀匹配, 很好的支持了无类路由。

[0097] 虽然以上描述了本发明的具体实施方式, 但是熟悉本技术领域的技术人员应当理解, 我们所描述的具体的实施例只是说明性的, 而不是用于对本发明的范围的限定, 熟悉本领域的技术人员在依照本发明的精神所作的等效的修饰以及变化, 都应当涵盖在本发明的权利要求所保护的范围内。

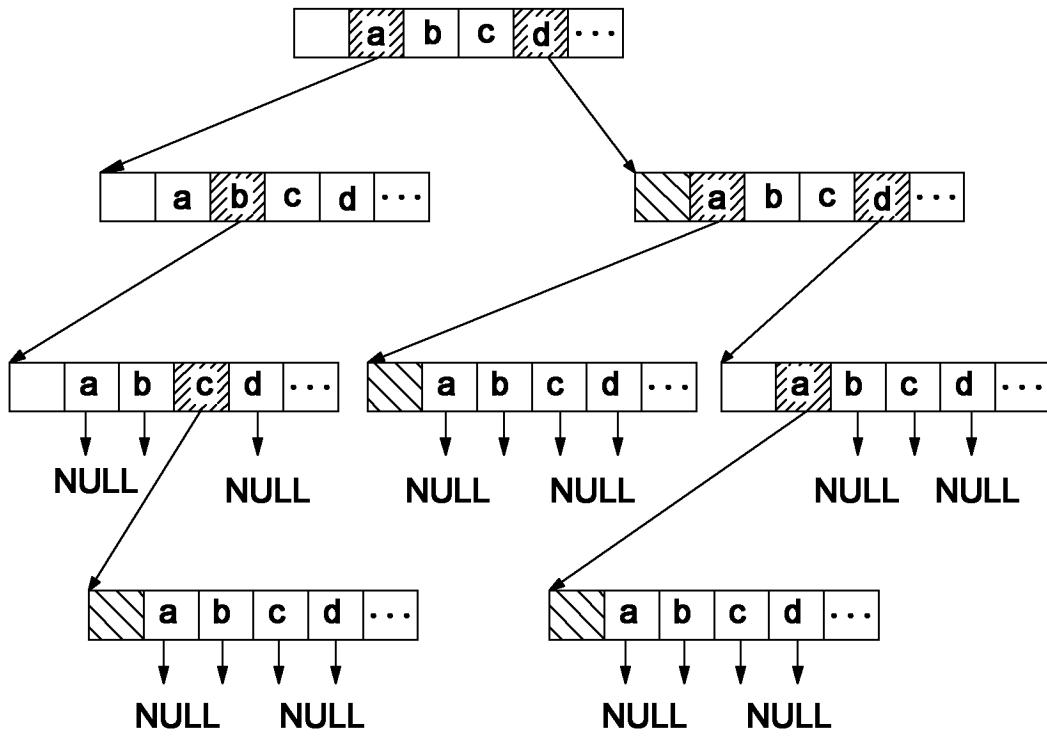


图 1

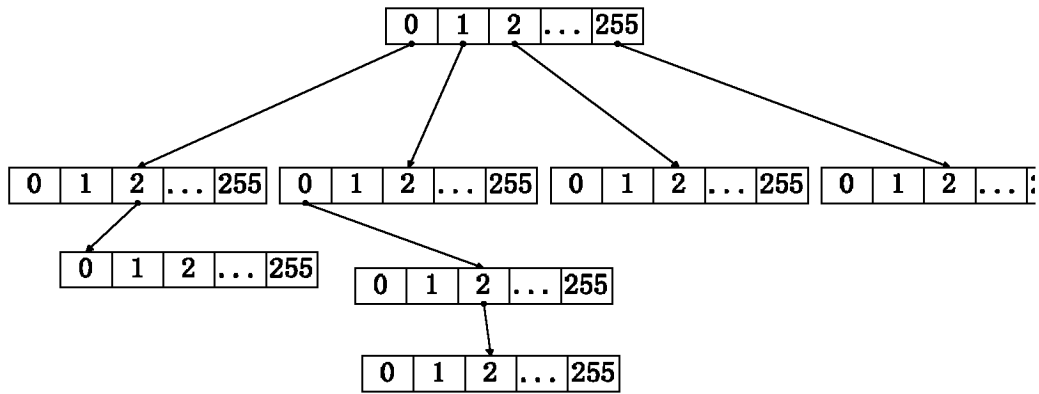


图 2

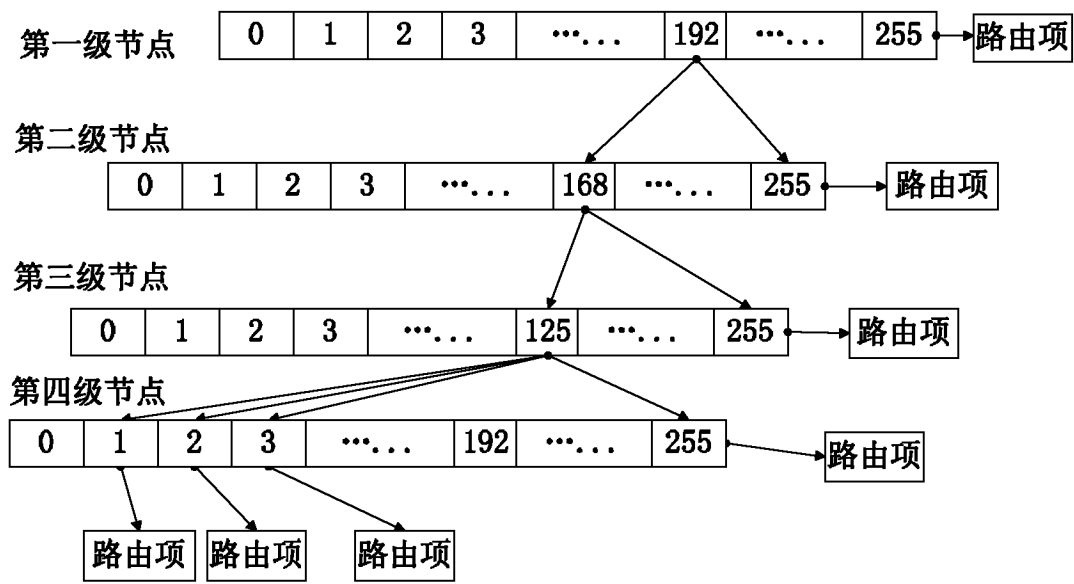


图 3

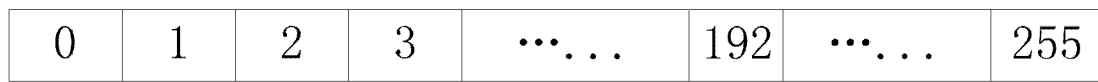


图 4

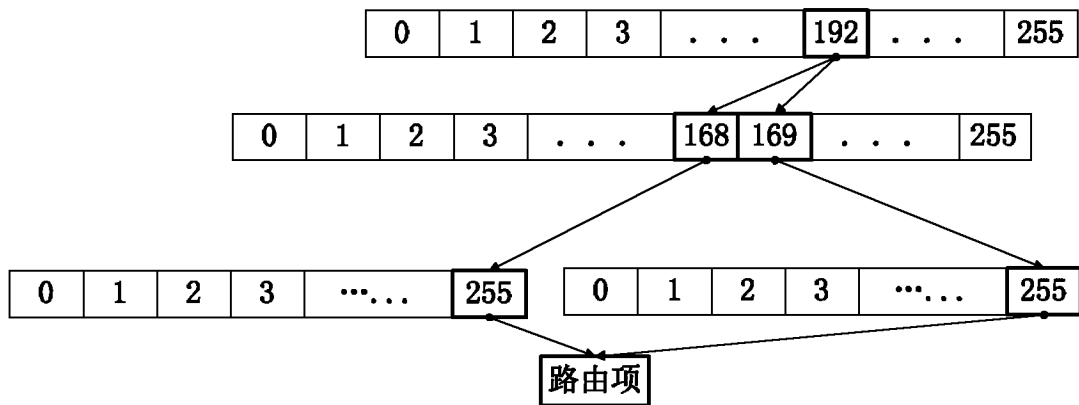


图 5

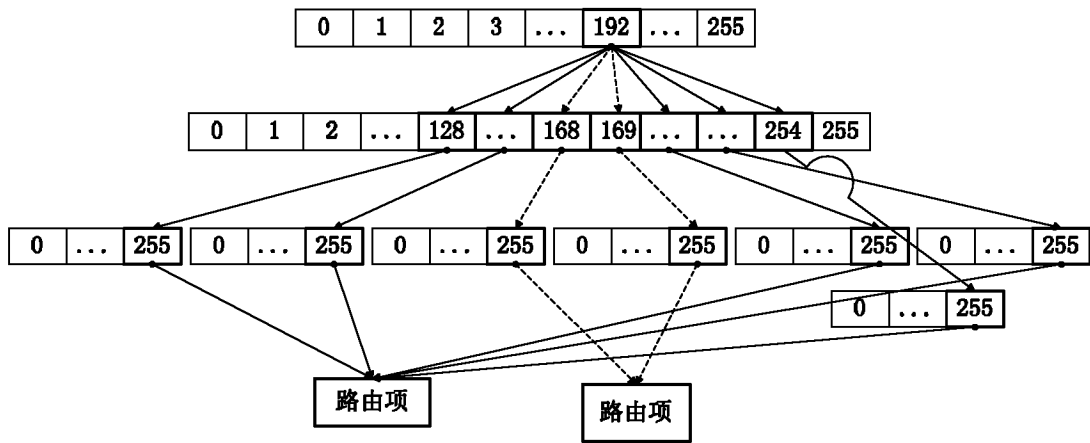


图 6

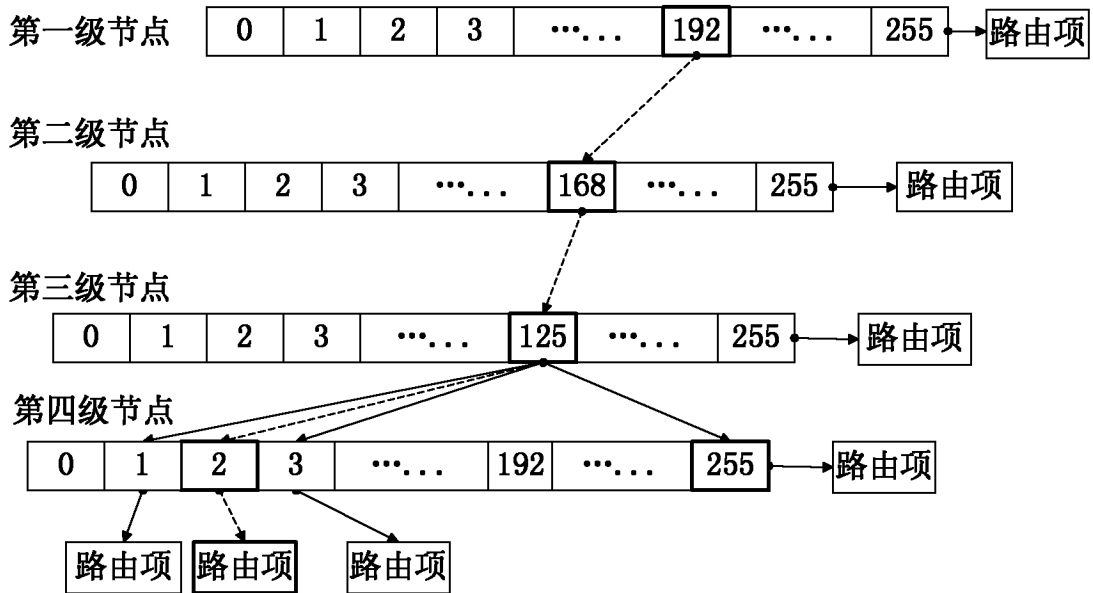


图 7

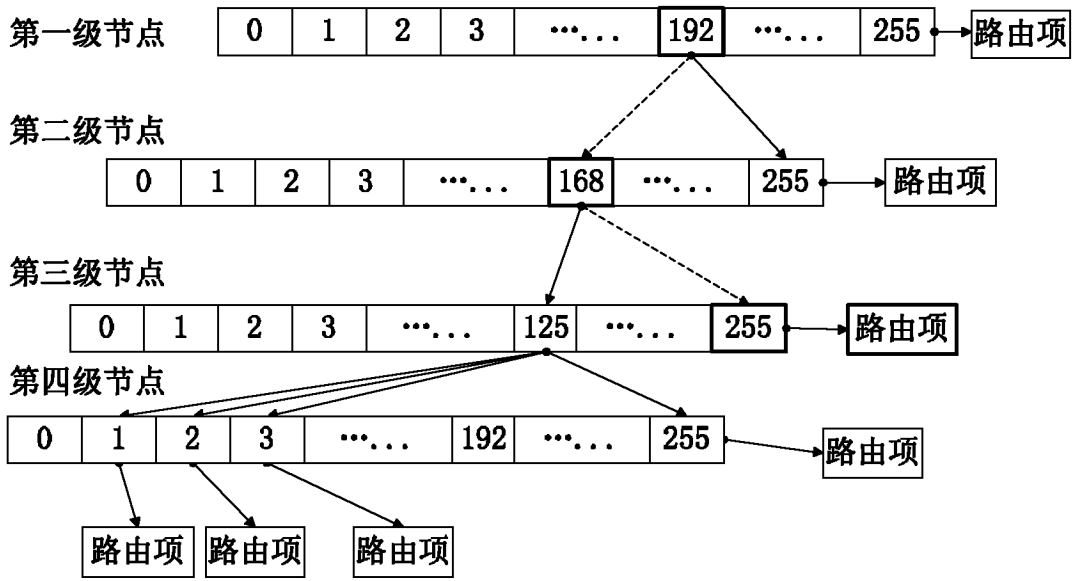


图 8

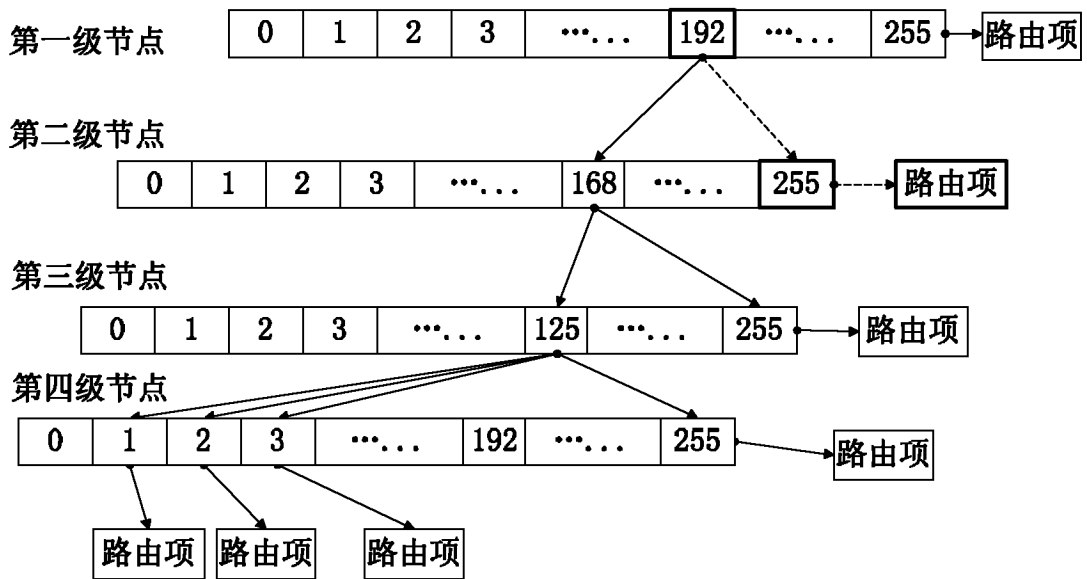


图 9